

Christoph Ußfeller

**Beiträge zur Lokalisation und zur modellbasierten
Lageregelung mobiler Roboter**

Beiträge zur Lokalisation und zur modellbasierten Lageregelung mobiler Roboter

Christoph Ußfeller



Universitätsverlag Ilmenau
2013

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Diese Arbeit hat der Fakultät für Maschinenbau der Technischen Universität Ilmenau als Dissertation vorgelegen.

Tag der Einreichung: 27. Februar 2012

1. Gutachter: Univ.-Prof. Dr.-Ing. habil. Mathias Weiß
(Technische Universität Ilmenau)

2. Gutachter: Prof. Dr. sc. nat. Dr.-Ing. Rudolf Baumgart-Schmitt
(Fachhochschule Schmalkalden)

3. Gutachter: Univ.-Prof. Dr.-Ing. habil. Klaus Zimmermann
(Technische Universität Ilmenau)

Tag der Verteidigung: 4. Juli 2012

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung

Verlagshaus Monsenstein und Vannerdat OHG

Am Hawerkamp 31

48155 Münster

www.mv-verlag.de

ISBN 978-3-86360-061-7 (Druckausgabe)

URN urn:nbn:de:gbv:ilm1-2012000489

Titelfoto: Veit Henkel | Fakultät für Maschinenbau, TU Ilmenau

Abstract

In the field of autonomous robotics the competitions in robot soccer have gained relevance as benchmark. The key aspect of these competitions is the accomplishment of coordinated behaviour of technical systems in a dynamical environment. This thesis presents contributions to the object localisation and position control of RoboCup Small Size League robots. A team of these robots is monitored and controlled externally by a top mounted vision system and a host.

The first part of this thesis covers implementation essentials of a segment oriented image processing system for object localisation. The image processing software yields high processing speed and robustness. The system utilises an externally provided static classifier. All pixel oriented operations (bayer interpolation, local gain, color classification, run length encoding) are integrated into a monolith to reduce memory usage. The classifier is generalising training data via approximated discrete voronoi decomposition of the RGB-colour space under different metrics. Labeled training data is required for the generation of this classifier. This data should be obtained by an algorithm which uses minimal prior knowledge. Therefore, the clustering algorithms DBSCAN and k-means and the feature extraction algorithm RANSAC are applied. The evaluation of these algorithms results in the necessity of user labeled data. The regression method Helmert transform is used for interpretation of the detected segments as object positions. The minimized error function is adapted for the context of image processing. This results in improved robustness of the system.

In the second part of this thesis a position control loop is introduced. The focus of the issue are the limits of system input and state. The limits of system input are caused by stiction/friction of rest and electrically producible torques. They are described by means of a multibody model of the robot. This description is transferred into a linear optimisation problem, which is solved with the simplex method. Thus as the robot's capability of acceleration as the appropriate configuration of drive forces is obtained. This model is utilised by velocity and position controllers. As the controlled robots must not collide, geometric constraints limit the system's state. This problem is coped by application of a path planner. The planner utilises the A*-algorithm to find the shortest path in a navigation graph. Variants of the navigation graph are implemented by a visibility graph and an adaptive rasterisation based on quadrees. For verification of the total system, the path planner is used in a reactive behaviour framework which enables the robot to score.

Kurzfassung

Auf dem Gebiet der autonomen Robotik haben Wettkämpfe im Roboterfußball Relevanz als Standardproblem gewonnen. Der Kernpunkt dieser Wettkämpfe besteht darin, die Aufgabe des koordinierten Verhaltens technischer Systeme in einer dynamischen Umgebung zu bewältigen. Die vorliegende Arbeit leistet Beiträge zur Objektlokalisierung und zur Lageregelung für Roboter der RoboCup Small Size League. Eine Mannschaft dieser Roboter wird durch eine über dem Spielfeld angebrachte Kamera und einen Leitrechner extern überwacht und gesteuert.

Im ersten Teil dieser Arbeit werden die Grundlagen des erstellten segmentorientierten Bildverarbeitungssystems für die Objektlokalisierung vorgestellt. Das Bildverarbeitungssystem zeichnet sich durch hohe Verarbeitungsgeschwindigkeit und Robustheit aus. Für die Segmentierung wird ein extern erzeugter statischer Farbklassifikator benutzt. Die Operationen auf der Ebene einzelner Bildelemente (Bayerinterpolation, lokale Verstärkung, Farbklassifikation, Lauflängenkodierung) sind zur effizienten Nutzung verfügbarer Speicherbandbreite monolithisch dargestellt. Der Aufbau des Farbklassifikators als Generalisierung von Trainingsdaten geschieht durch eine approximierte diskrete Voronoidekomposition des RGB-Farbraumes unter Nutzung der Mahalanobisdistanz bzw. eines minimalen Stichprobenabstandes. Für den Aufbau des Klassifikators sind zugeordnete Trainingsdaten erforderlich. Diese sollen durch ein Verfahren mit minimalen Vorwissen aus dem Videostrom gewonnen werden. Untersucht werden hierzu DBSCAN und k-means als Algorithmen der Clusteranalyse sowie RANSAC als Verfahren der Merkmalsextraktion. Die Untersuchung dieser Verfahren ergibt, dass der Nutzer zugeordnete Trainingsdaten bereitstellen muss. Zur Interpretation der detektierten Segmente als Objektpositionen wird das Regressionsverfahren der Helmert-Transformation genutzt. Hierbei wird die zu minimierende Fehlerfunktion für den Kontext der Bildverarbeitung adaptiert, woraus eine gesteigerte Robustheit des Gesamtsystems resultiert.

Im zweiten Teil der Arbeit wird ein Lageregelkreis realisiert. Den Schwerpunkt dieses Teils stellen die Begrenzungen von Stellgröße und Zustand des Roboters dar. Die Begrenzungen der Stellgröße werden durch die Haftreibungsgrenze und die von den eingesetzten Elektromotoren erzeugbaren Momenten verursacht. Sie werden mit Hilfe eines Mehrkörpermodells beschrieben. Diese Beschreibung wird in ein lineares Optimierungsproblem überführt und durch das Simplex-Verfahren gelöst. Auf diese Weise ergeben sich sowohl die Beschleunigungsfähigkeit des Roboters, als auch die hierfür notwendige Antriebskraftkonfiguration. Dieses Modell wird in einem Geschwindigkeits- und zwei Lagereglern verwendet. Da die Roboter im Einsatz nicht kollidieren dürfen, beschränken geometrische Nebenbedingungen den Roboterzustand. Auf dieses Problem wird durch die

Anwendung eines Pfadplaners eingegangen. Der Pfadplaner basiert auf dem A*-Algorithmus und sucht nach dem kürzesten Weg in einem Navigationsgraphen. Varianten zur Erzeugung des Navigationsgraphen werden mit einem Sichtbarkeitsgraphen und einer adaptive Rasterung durch einen Quadtree bereitgestellt. Als Funktionsnachweis des Gesamtsystems wird der Pfadplaner in einer reaktiven Struktur eingesetzt. Hiermit werden erfolgreich Torschüsse ausgeführt.

Danksagung

Ich möchte mich herzlich bei meinen Förderern Herrn Univ.-Prof. Dr.-Ing. habil. Mathias Weiß, Herrn Univ.-Prof. Dr.-Ing. habil. Klaus Zimmermann, Frau Dr.-Ing. Marion Braunschweig und Thorsten Rieß für ihre Unterstützung und konstruktive Kritik bedanken. Danke dafür, dass mir hier in den Fachgebieten Rechneranwendung im Maschinenbau und Technische Mechanik der Technischen Universität Ilmenau viele Jahre lang ein Umfeld für freie, kreative Arbeit und intensive Betreuung geboten wurde.

Wesentlichen Einfluß auf das Gelingen dieser Arbeit hatten Michael Jahn, Steffen Lerm, Peter Landsmann, Florian Schale, Andreas Müller, Ricardo Dill, Sylvia Schwachenwalde, Tobias Kästner, Tobias Linke, Hans-Peter Walkling, Oliver Birli, Isabel Husung sowie Herr Dr.-Ing. Sebastian Frank, Herr Dr.-Ing. Bernd Fiedler und Herr Dr.-Ing. Erik Gerlach durch Vorarbeiten, technische Rahmensetzung sowie offenen Dialog. Ich bedanke mich für die gute Zusammenarbeit.

Mein Dank für ausgiebigen fachlichen Diskurs und wertvolle Anregungen zu dieser Arbeit gilt vor allen Daniel Heß, Carsten Behn, Aljona Chervakova, Norbert Thieme, Felix Becker, Jana Popp, Thorsten Hüfner, Prof. Dr. rer. nat. habil. Joachim Steigenberger, Jens Schreyer, Lars Hartmann, Tobias Schmidt, Lutz Heermann und Manuel Schöne.

Ich danke meinen Freunden in prägenden, aufregenden und lehrreichen Zeiten, vor allem Andreas Engelbert, Ingo Gryl, Frank Beinersdorf, Sebastian Fett, Volker Henze, Christian Döring, Toini Schmitz, Caecilie von Teichmann und Christian Cermann. Für kritische, destruktive und erfrischende Diskussionen danke ich besonders Enrico Schmidtke und Simone Greiner-Fuchs. Ohne Euch wäre diese Arbeit nicht entstanden.

Ich möchte mich bei Frau Claudia Umlauf und Herrn Manfred Mohaupt herzlich bedanken, die durch das Fördern meiner Kreativität und Neugier wesentlichen Einfluß auf meine Entwicklung hatten.

Den größten Anteil daran, dass ich heute diese Arbeit vorlegen kann, haben *Annette, Walter, Herbert, Brigitta, Fritz, Hilde* und *Constanze*. Ihr habt mich in Liebe großgezogen, Ihr habt mir immer vertraut und mich unterstützt. Danke!

Inhaltsverzeichnis

1. Einleitung	1
1.1. RoboCup an der TU Ilmenau	1
1.2. Präzisierte Aufgabenstellung	2
1.3. Realisierte Softwaremodule	6
1.4. Modellierung des Verhaltens	8
1.4.1. Paradigmen der Verhaltensmodellierung	8
1.4.2. Realisierung des Mannschaftsverhaltens	10
I. Objektlokalisierung durch digitale Bildverarbeitung	13
2. Subsymbolische Verfahren der Bildverarbeitung	15
2.1. Vorverarbeitung	16
2.1.1. Rohdatenaufbereitung durch Bayerinterpolation	16
2.1.2. Helligkeitsausgleich	21
2.2. Farbklassifikation	24
2.2.1. Klassifikation durch Generalisierung von Stichproben	28
2.2.2. Autonome Stichprobenerhebung	34
2.3. Segmentierung	45
2.3.1. Lauflängenkodierung	46
2.3.2. Segmentvereinigung	47
3. Symbolische Auswertung – Bildinterpretation	51
3.1. Trikots	51
3.1.1. Anforderungen	51
3.1.2. Kodierung	51
3.1.3. Festlegung eines Trikots	52
3.2. Lokalisation	55
3.2.1. Helmert-Transformation	55
3.2.2. Helmert-Transformation mit Wichtungen	57
3.2.3. Vergleich verschiedener Gütekriterien	62

II. Steuerung eines RoboCup-Spielers	71
4. Regelung eines omnidirektionalen Antriebssystems	73
4.1. Modellierung eines omnidirektional getriebenen Roboters als Mehrkörpersystem	77
4.1.1. Koordinatensysteme	79
4.1.2. Roboterzustand	80
4.1.3. Schnittgrößen	81
4.1.4. Dynamik des Omnidirektionalrades	82
4.1.5. Dynamik des Antriebes	92
4.1.6. Dynamik des Grundkörpers	96
4.2. Beschränkung der Beschleunigungsfähigkeit eines omnidirektionalen Antriebssystems	102
4.2.1. Umstrukturierung von Impuls- und Drehimpulssatz des Grundkörpers	103
4.2.2. Projektionen der Radwinkelgeschwindigkeiten, der Radgeschwindigkeiten und der Schnittmomente	104
4.2.3. Approximation der Haftreibungsgrenze	105
4.2.4. Begrenzung durch realisierbare Antriebsmomente	106
4.2.5. Formulierung der Beschleunigungsfähigkeit als lineares Optimierungsproblem	108
4.2.6. Erweiterung des Modells um den Radschlupf	114
4.2.7. Vereinfachung für dreirädrige Roboter	117
4.3. Regelung	119
4.3.1. Geschwindigkeitsregler	119
4.3.2. Lageregler	120
5. Pfadplanung	131
5.1. Graphensuche	133
5.1.1. Definitionen	133
5.1.2. A*-Algorithmus	134
5.2. Aufbau des Navigationsgraphen als Sichtbarkeitsgraph	137
5.2.1. Approximation der Hindernisse durch Polygone	139
5.2.2. Punkt-im-Polygon-Test	141
5.2.3. Strecke-schneidet-Polygon-Test	141
5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition	143
5.3.1. Eigenschaften einer Zelle	144
5.3.2. Adaptive Zelldekomposition durch Aufbau eines Quadtree	146
5.3.3. Vergleich zwischen adaptiver Zelldekomposition und konstruiertem Sichtbarkeitsgraph	150

5.4. Anwendung der Pfadplanung	154
5.4.1. Nachbearbeitung eines Pfades	154
5.4.2. Realisierte Manöver	155
6. Zusammenfassung und Ausblick	159
A. Verwendung des Computeralgebrasystems Maxima	165
B. Matrizen des Mechanikmodells	169
C. Matrizen des reduzierten Mechanikmodells	177
Symbolverzeichnis	xiii
Abbildungsverzeichnis	xxv
Tabellenverzeichnis	xxvii
Liste der Algorithmen	xxix
Literaturverzeichnis	xxxix

1. Einleitung

1.1. RoboCup an der TU Ilmenau

Der RoboCup [rob] ist eine Initiative zur Förderung der Erforschung künstlicher Intelligenz. Hierbei treten Wissenschaftler und Studenten zum Wettkampf im Fußball autonomer technischer Systeme an. Der emotionsgeladene Sport ist dabei einerseits erfolgreicher Attraktor wissenschaftlichen Interesses, bietet andererseits durch die Standardisierung der Rahmenbedingungen die Möglichkeit der Durchführung fairer Wettkämpfe. Der Fußball ist durch den RoboCup zu einem neuen Standardproblem der künstlichen Intelligenz geworden. Er fokussiert sowohl auf die physischen Aspekte der Robotik, als auch auf Methoden zur Darstellung autonomen Verhaltens technischer Systeme. Der künstliche Spieler muss in der Lage sein, in einer sich schnell verändernden und nicht vollständig oder exakt wahrnehmbaren Umgebung zielgerichtete Handlungen auszuführen. Aufgrund der Breite des Spektrums zu bewältigender Teilaufgaben ist der RoboCup in Ligen mit unterschiedlichen Schwerpunkten unterteilt: Die *Simulations League* abstrahiert weitgehend von den physischen Aspekten und widmet sich ausschließlich der Modellierung des Verhaltens. In der *Standard Platform League* werden physische Spieler gleicher Beschaffenheit benutzt. Somit können Probleme der Interaktion mit einer realen Umgebung bewältigt werden, ohne einen leistungsfähigen Spieler entwickeln zu müssen. Dagegen ist in den Ligen *Small Size*, *Middle Size* und *Humanoid* die Entwicklung der Spieler Teil der Forschung. In der *Small Size League* werden die Spieler durch eine externe Bildverarbeitung detektiert und extern gesteuert. In der *Middle Size League* verfügen die Spieler ausschließlich über eigene Sensorik und interne Datenverarbeitung. In der *Humanoid League* müssen die Spieler das Fußballspiel durch zweibeinigen Gang bewältigen.

Das ambitionierte Ziel der Initiative ist es, im Jahr 2050 die Weltmeisterschaft des Menschen-Fußball zu schlagen.

Studentischer Einsatz führte zu einem Engagement der Fachgebiete Rechneranwendung im Maschinenbau und Technische Mechanik der Fakultät für Maschinenbau der Technischen Universität Ilmenau im Bereich des RoboCup. Seit 1999 werden hier vielfältige Aspekte des Ingenieursstudiums im Rahmen der RoboCup Small Size League vereint. Im Fokus des Interesses steht die Gestaltung eines erfolgreichen Spielers. Hierfür müssen Teilaufgaben in den Bereichen Kon-

1. Einleitung

struktion, Antriebstechnik, Elektronik, Sensorik und Regelungstechnik bewältigt werden. Das RoboCup-Projekt ist ein wichtiger Teil der Ausbildung, in dem Fragestellungen zu aktuellen Forschungsinteressen bearbeitet werden. Hieraus resultierten Arbeiten zur Konstruktion von Spielern [Abr01, Jah09, Sch09, Mül09] und zur Modellbildung [Lan05, Füc06, ZWBR07, ZZB09, ESS10]. Weitere Schwerpunkte bestehen in Arbeiten zu Systemintegration [Röl00, Jah05, Lan05, JWZ⁺08, Ußf08, Jah09, Pet10, Sch10, FSU11, Dil11, Sch11] und Regelung mobiler Roboter [Lan05, Jah05, ZWBR07, Ußf07, Ußf08, Jah09, Sch10] sowie Objektlokalisation durch digitale Bildverarbeitung [Röl00, Ler08, Ußf11].

1.2. Präzisierte Aufgabenstellung

Ausgangslage Der Kontext dieser Arbeit ist eine Spielumgebung für das Fußballspiel autonomer Roboter der RoboCup Small Size League. Teil dieser Spielumgebung sind 5 omnidirektionale Roboter des Modells „Lukas“ nach [Jah09], siehe Abbildungen 1.1 und 1.2. Diese Roboter haben einen Umkreis von 180 mm Durchmesser, ihre Masse beträgt 1,7 kg. Sie werden elektrisch angetrieben und erreichen Geschwindigkeiten über 4 ms^{-1} . Diese Spieler verfügen über Lage-regler [Lan05], welche es ihnen erlauben, selbstständig eine Position auf dem Spielfeld anzufahren. Die Lage der beteiligten Objekte wird mit Hilfe einer über dem Spielfeld angebrachten Kamera detektiert. Hierfür wird ein segmentbasiertes Bildverarbeitungssystem nach [Ler08] verwendet. Anhand der Situation auf dem Spielfeld werden in einem Strategiemodul Befehle für die einzelnen Spieler generiert und an die Spieler gesendet. Die Bildverarbeitung und die Strategie werden auf einem Leitrechner ausgeführt.

Schwachpunkte des aktuellen Systems Das eingesetzte Bildverarbeitungssystem segmentiert die einzelnen Bilder des Videostroms und wertet die gefundenen Segmente zur Lagebestimmung aus. Die Segmentierung geschieht mit Hilfe einer Zuordnung der einzelnen Bildelemente zu Farbklassen. Die in diesem System eingesetzte Kamera bietet eine Auflösung von 768x576 Bildelementen und arbeitet nach dem Zeilensprungverfahren. Durch diese niedrige Auflösung bedingt, stehen für die Lagebestimmung eines Objektes nur wenige Bildelemente zur Verfügung. Damit haben zufällige Störungen bei der Bildaufnahme großen Einfluss auf die detektierte Lage der einzelnen Objekte. Insbesondere die Bestimmung der Orientierung hat sich als Schwachpunkt herausgestellt. Da die einzelnen Spieler über Lagebefehle bezüglich ihres eigenen Koordinatensystems

¹Das aktuelle Reglement legt eine Spielfeldgröße von ca 6m x 4m fest. Das Spielfeld des RoboCup Versuchsstandes der TU Ilmenau bietet aufgrund begrenzter räumlicher Ressourcen eine Spielfläche von 2,74m x 1.52m

1.2. Präzisierte Aufgabenstellung

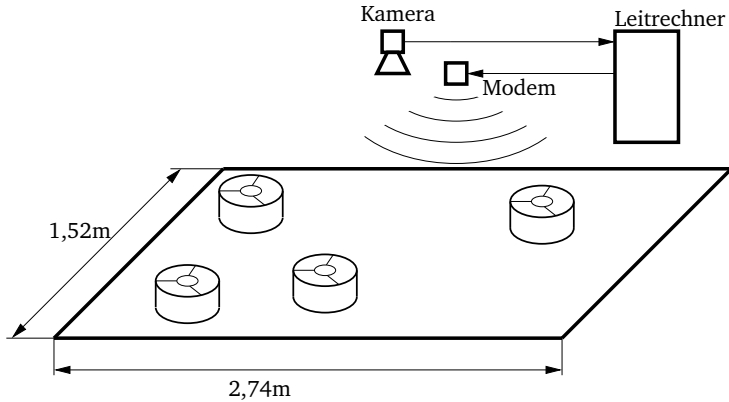


Abbildung 1.1.: Reale Spielumgebung mit Spielfeld¹, externer Kamera, Leitrechner und omnidirektionalen Robotern sowie Funkanbindung

angesteuert werden, haben gerade Orientierungsfehler großen Einfluss auf die Regelgüte des Gesamtsystems. Hieraus ergibt sich die Forderung nach einer Verbesserung der Genauigkeit der Lagebestimmung.

Die Spieler des Modells Lukas zeigen durch ihre Radanordnung und Antriebsauslegung eine starke Begünstigung der Längsdynamik zulasten der Querdynamik, siehe Abbildung 1.2. Diese Asymmetrie erschwert die Durchführung präziser Zustellbewegungen. Die Antriebsstränge für die Längsrichtung enthalten Kegelradgetriebe, welche sich in der Praxis als sehr verschleißbehaftet erwiesen haben. Deshalb wurde in [Mül09, Sch10] ein Nachfolgemodell mit Direktantrieb und einer weniger stark ausgeprägten Asymmetrie entwickelt, siehe Abbildung 1.3. Der Schussmechanismus basiert auf einer Feder als Energiespeicher. Die Energieversorgung des Spielers geschieht durch Lithium-Eisen-Phosphat-Akkumulatoren, welche in ein Schnellwechselsystem integriert sind. Von diesem Modell existiert ein noch nicht vollständig fertiggestellter Prototyp. Leider kann der vorhandene Lageregler nicht ohne weiteres für das neue Modell angepasst werden, da die Implementierung sehr stark an der speziellen Radanordnung des Vorgängers orientiert ist. Deshalb besteht die Forderung nach einem allgemeineren Modell, welches leicht auf zukünftige Änderungen angepasst werden kann.

1. Einleitung

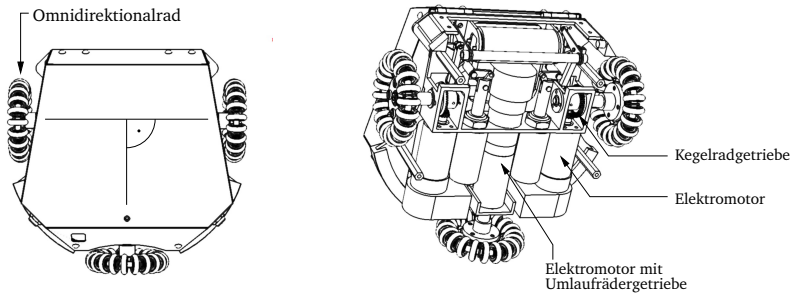


Abbildung 1.2.: Modell Lukas[Jah09]

Lösungsansätze Als Lösungsansätze für die Probleme der Objektllokalisierung werden zwei Maßnahmen in Betracht gezogen: Die erste Maßnahme ist es, die Anzahl der zur Lagebestimmung verwendbaren Bildelemente zu erhöhen. Dies wird einerseits durch das Verwenden einer Kamera höherer Auflösung realisiert. Hiermit sind gesteigerte Anforderungen an die Verarbeitungsgeschwindigkeit des Bildverarbeitungssystems verbunden. Andererseits kann durch eine Umgestaltung des zu erkennenden Spielmusters eine bessere Ausnutzung der verfügbaren Fläche erreicht werden. Hierbei ergeben sich jedoch gesteigerte Anforderungen an die Farbklassifikation, da sich die Menge der zu unterscheidenden Farben vergrößert. Die zweite Maßnahme zur Verbesserung der Leistungsfähigkeit des Bildverarbeitungssystems betrifft die Auswertung der erkannten Segmente. Mit Hilfe eines Verfahrens der Ausgleichsrechnung können erkannte Segmente optimal in das zu detektierende Spielmuster eingepasst werden. Hierdurch wird die Lage eines Spielers mit den zur Verfügung stehenden Informationen bestmöglich erkannt.

Der Lageregler hat die Aufgabe, den Spieler in eine Solllage zu überführen. Die positiven Erfahrungen mit dem bisherigen modellbasierten Lageregler sind Motivation dafür, das zugrunde liegende Modell zu verallgemeinern und das Regelungskonzept auf den neuen Spieler zu übertragen. Ein wichtiger Aspekt bei der Überführung des Spielers in eine Solllage ist jedoch die Existenz von Hindernissen. Hieraus entsteht das Problem, geeignete Führungsgrößen des Lagereglers zu generieren, um eine kollisionsfreie Bewegung sicherzustellen.

Lösungsstruktur Hiermit ergibt sich die Gliederung dieser Arbeit in zwei Teile. Im ersten Teil wird ein neues Bildverarbeitungssystem zur Lokalisation der einzelnen Objekte der Spielumgebung erarbeitet. Kapitel 2 widmet sich der Seg-

1.2. Präzisierte Aufgabenstellung

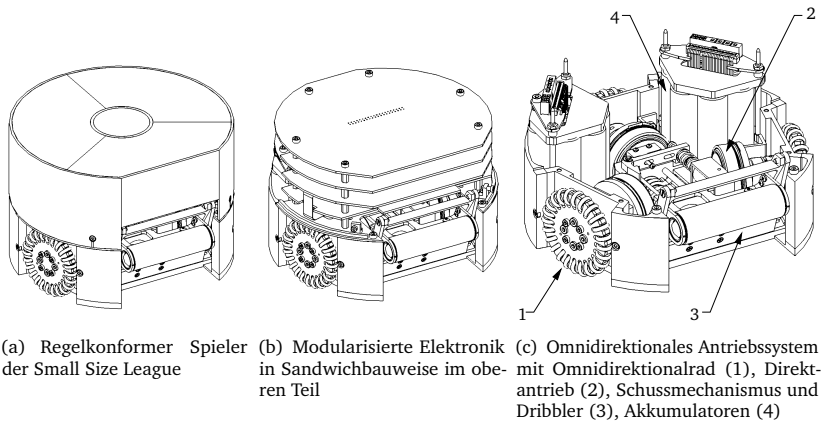


Abbildung 1.3.: RoboCup Spieler der TU Ilmenau, Modell M11 [Mül09, Sch10]

mentierung der Einzelbilder des Videostroms. Ein Schwerpunkt dieses Kapitels ist dabei das Teilproblem der Farbklassifikation, welche die Grundlage erfolgreicher Segmentierung ist. In Kapitel 3 wird ein neues Spielmuster festgelegt, um die für Markierungen verfügbare Fläche besser auszunutzen. Außerdem wird das Lokalisationsverfahren der Helmert-Transformation benutzt, um aus den detektierten Segmenten auf die Spielerlagen zu schließen. Das Verfahren wird modifiziert, um Zusatzwissen über die erwartete Objektgeometrie auswerten zu können. Diese Modifikation wird mit einem synthetischen Test verifiziert.

Der zweite Teil wendet sich dem Problem der Überführung des Spielers in eine Solllage zu. Hierfür wird im Kapitel 4 eine modellbasierte Lageregelung realisiert. Den Schwerpunkt dieses Kapitels bildet dabei die Erarbeitung eines verallgemeinerten Modells zur Beschreibung der Beschleunigungsfähigkeit eines omnidirektionalen Antriebssystems. Zur Sicherstellung kollisionsfreier Bewegung werden im Kapitel 5 zwei kombinatorische Pfadplanungsverfahren angewendet.

Um die Funktionsweise der erarbeiteten Lösungen zu demonstrieren, werden diese in eine Softwarestruktur integriert. Mit Hilfe dieser Softwarestruktur werden sowohl die Spielerverhalten Torschuss und Verteidigung, als auch ein einfaches Mannschaftsverhalten realisiert.

1. Einleitung

1.3. Realisierte Softwaremodule

Im Rahmen vorliegender Arbeit wurden einzelne Softwaremodule zur Ansteuerung einer RoboCup-Mannschaft implementiert. Die zentralen Module der Umgebungen und der Strategie sowie das Modul der Visualisierung bilden ein verteiltes System. Dieses ist in Abbildung 1.4 dargestellt. Die realisierte Struktur basiert auf [Ußf08] und erweitert diese um Möglichkeiten der Visualisierung aus dem Strategiemodul heraus. Die implementierten Module werden im Folgenden kurz umrissen.

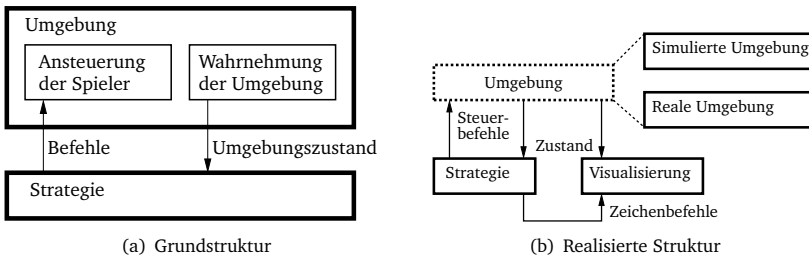


Abbildung 1.4.: Softwaremodule des autonomen Fußballspiels

Umgebung Die Umgebung bildet den Rahmen der Handlungen der autonomen Roboter. Das Softwaremodul der Umgebung stellt dem Modul der Mannschaftsstrategie einerseits Informationen über den Zustand der abgebildeten Umgebung durch Ereignisse und andererseits Handlungsmöglichkeiten in der Umgebung durch Steuerbefehle zur Verfügung.

Reale Umgebung Das Softwaremodul der realen Umgebung bindet die zu steuernden Roboter durch Funkkommunikation an. Durch digitale Bildverarbeitung (siehe Kapitel 2 und 3) wird der Zustand der Umgebung detektiert.

Simulierte Umgebung Der Fokus der realisierten Simulationsumgebung liegt auf der Nachbildung der Spielerbewegungen. Hierzu werden die Modelle nach [Lan05] bzw. Kapitel 4 verwendet. Hierdurch können einerseits die Auswirkung von elektrischen und mechanischen Eigenschaften sowie Reglerparametern überprüft werden. Andererseits ist die Simulationsumgebung das wichtigste Hilfsmittel der Verhaltensmodellierung. Der Umgebungszustand kann dem Strategiemodul wahlweise idealisiert oder rauschbehaftet bereitgestellt werden.

Aspekte wie die Bewegung des Balls oder Kollisionen zwischen Objekten wurden nur rudimentär behandelt.

Strategie Im Modul der Mannschaftsstrategie werden die Steuerbefehle für die einzelnen Spieler generiert. Die Einbettung in die Umgebung geschieht transparent – für ein Strategiemodul ist nicht erkennbar, ob die beeinflusste Umgebung real oder simuliert ist. Das Verhalten der Mannschaft wird durch eine hybrid-deliberative (siehe Abschnitt 1.4.2) Struktur realisiert.

Visualisierung Die Visualisierung bildet eine graphische Schnittstelle zwischen der Simulationsumgebung bzw. der Mannschaftsstrategie und dem Benutzer. Sie stellt die einzelnen Objekte des Spieles dar und bietet dem Strategiemodul eine Zeichenfläche sowie einen Satz einfacher Zeichenbefehle. Diese Möglichkeit macht das Visualisierungsmodul zu einem wichtigen Entwicklungswerkzeug, da es den Prozess der Fehlersuche bei der Implementierung der Roboterverhalten erheblich beschleunigt.

Weiterhin werden Hilfsprogramme für das Modul der realen Umgebung bereitgestellt. Um die digitale Bildverarbeitung in Betrieb zu nehmen, werden folgende Module benutzt:

Graphische Nutzerschnittstelle für das Einrichten der Bildverarbeitung

Die graphische Nutzerschnittstelle für das Einrichten der Bildverarbeitung wird zur Kamerajustierung und für die Bestimmung des lokalen Helligkeitsausgleiches (siehe Abschnitt 2.1.2) genutzt. Ihr Haupteinsatzzweck ist die Bereitstellung von Trainingsdaten für das Farbenlernen nach Abschnitt 2.2.

Klassifikatoraufbau Für die Abbildung des RGB-Farbraumes auf die Menge markierender Farben wird in Abschnitt 2.2 eine diskrete Voronoidekomposition anhand verschiedener Kriterien durchgeführt. Diese Dekomposition wird als Klassifikator für die digitale Bildverarbeitung benutzt und ist die Grundlage der realisierten segmentbasierten Farbbildauswertung. Die Module zum Klassifikatoraufbau verarbeiten vom Nutzer bereitgestellte Stichproben der Farbklassen und liefern Dekompositionen des Farbraumes in Form eines einfachen Austauschformates.

Hinweise zur Implementierung Die Module der realen und simulierten Umgebung, der Strategie, der graphischen Nutzerschnittstelle für das Einrichten der Bildverarbeitung sowie des Klassifikatoraufbaus sind in der C++-

1. Einleitung

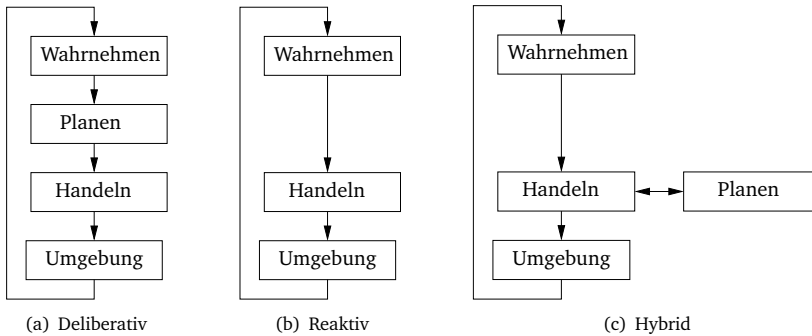


Abbildung 1.5.: Paradigmen der Verhaltensmodellierung

Programmiersprache [Str00] verfasst. Hierbei werden Teilaufgaben der Nebenläufigkeit und der Deallokation dynamisch allozierter Ressourcen mit Hilfe der Boost-Bibliothek [Boo] gelöst. Sie werden durch den C++-Compiler der Gnu Compiler Collection [gcc] in maschinell ausführbare Programme übersetzt. Die Module der realen Umgebung und der graphischen Nutzerschnittstelle für das Einrichten der Bildverarbeitung verwenden eine Bibliothek des Herstellers [pro] zur Kommunikation mit der Kamera. Die graphische Nutzerschnittstelle ist mit Hilfe der gtkmm-Bibliothek [gtk] realisiert.

Die Module der Umgebungen, der Strategie und der Visualisierung sind mit Hilfe der Common Object Request Broker Architecture [cor] gekoppelt. Die C++-Module nutzen hierfür omniORB [omn]. Die Visualisierung ist in der Java-Programmiersprache [jav] verfasst.

1.4. Modellierung des Verhaltens

1.4.1. Paradigmen der Verhaltensmodellierung

Es existieren viele Strategien, einem autonomen technischen System erwünschtes Verhalten aufzuprägen. Diese lassen sich jedoch grob den drei Kategorien deliberativ, reaktiv und hybrid deliberativ-reaktiv zuordnen, siehe Abbildung 1.5.

Deliberatives Paradigma Beim deliberativen Ansatz [Nil69] werden Information über den Umgebungszustand durch Sensoren aufgenommen und in ein

Modell der Umgebung integriert. Das Umgebungsmodell ermöglicht die Planung von Handlungssequenzen. Eine nicht direkt lösbare Aufgabe kann so durch die Abarbeitung einer Folge von Teilhandlungen gelöst werden. Durch die Ausführung einer Teilhandlung verändert das autonome System den Zustand der Umgebung.

Reaktives Paradigma Die Voraussetzung eines robusten Umgebungsmodells führt zu starker Einschränkung der Anwendbarkeit des deliberativen Paradigmas für praktische Aufgaben in komplexen Umgebungen. Durch den Verzicht auf ein Umgebungsmodell versuchen die Vertreter des reaktiven Paradigmas [Bro90, JC98] robuste autonome Systeme zu realisieren. Komplexe Verhaltensweisen entstehen durch situationsabhängige Überlagerung einfacher Teilverhalten.

Hybrid deliberativ-reaktives Paradigma Dem Vorteil des deliberativen Paradigmas, nicht direkt lösbare Aufgaben in Sequenzen lösbarer Teilaufgaben zu überführen steht der Nachteil der Nichtanwendbarkeit in komplexen Umgebungen entgegen². Dem Vorteil des reaktiven Paradigmas, kein explizites Umgebungsmodell zu benötigen, steht der Nachteil der Unvollständigkeit entgegen. Im reaktiven Paradigma kann das technische System bei der Bewältigung einer Aufgabe scheitern, obwohl eine Lösung existiert.

Das hybrid deliberativ-reaktive Paradigma vereint die Vorzüge des deliberativen und des reaktiven Paradigmas. In einem reaktiven Kontext werden Teilaufgaben, für die ein ausreichend robustes Modell³ bereit steht, durch geplante Handlungssequenzen bewältigt. Hybride Architekturen haben im autonomen Roboterfußball eine große Bedeutung[BBBV05]. Während es schwierig ist, ein explizites, global gültiges Modell des Fußballspiels zu erarbeiten, können Teilaufgaben durch geplante Handlungssequenzen gelöst werden. Ein Beispiel hierzu ist die Überführung eines Roboters in eine Ziellage durch geometrische Pfadplanung.

1. Einleitung

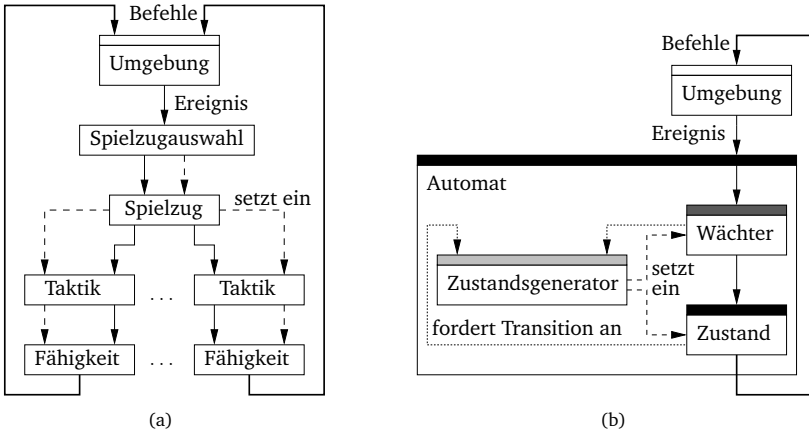


Abbildung 1.6.: Verhaltensmodellierung im RoboCup durch Struktur mit festgelegter Hierarchie nach [BBBV05] (a) sowie realisierte kaskadierbare Struktur (b)

1.4.2. Realisierung des Mannschaftsverhaltens

Das Mannschaftsverhalten wird durch eine hybride Struktur erzeugt. Die Struktur ist angelehnt an [BBBV05] (siehe Abbildung 1.6(a)), jedoch stark vereinfacht. Die Begründung dieser Vereinfachungen liegt darin, dass es das primäre Ziel des Softwaremoduls zur Mannschaftssteuerung ist, die Funktionsfähigkeit der einzelnen Funktionseinheiten (Bildverarbeitung, Lageregelung, Pfadplanung) nachzuweisen. Da die Struktur kaskadierbar ist, kann eine feinere Anpassung des Abstraktionsgrades zur Lösung von Teilaufgaben erreicht werden, als bei der festen Aufteilung in primitive, taktische und strategische Ebene in [BBBV05]. Die Struktur reagiert durch Auswertung eines Regelwerkes auf den Zustand der Umgebung. Das Regelwerk ist in einem Verbund endlicher

²Ein breites Feld von Lösungen des Problems der Verfügbarkeit eines explizit formulierten Umgebungsmodells besteht in der Verwendung von Lernverfahren. Diese Verfahren sollen es dem technischen System ermöglichen, selbst ein Umgebungsmodell zu entwickeln und dieses durch die Überprüfung von Hypothesen zu verbessern. Beispiele sind die Bestrebungen, technische Systeme ein eigenes Sprachsystem entwickeln zu lassen [OEW99, SV97] sowie die Verwendung bestärkender Lernverfahren [TP97, SB98] in Anwendungen von Navigationsaufgaben bis zum Backgammon. Durch die erlernten Umgebungsmodelle sind diese Systeme zu deliberativer Handlung fähig. Problematisch sind jedoch die Komplexität beherrschbarer Umgebungen sowie die Effizienz der Lernverfahren bezüglich der Mächtigkeit verfügbarer Trainingsdaten.

³Dieses Modell kann dabei explizit formuliert sein oder durch Lernverfahren ermittelt werden.

1.4. Modellierung des Verhaltens

Automaten kodiert, siehe Abbildung 1.6(b). Dieser Verbund ist objektorientiert modelliert. Die wichtigsten Datentypen des Entwurfes sind das Ereignis, der Zustand, der Wächter und der Zustandsgenerator. Dabei sind die abstrakten⁴ Metadatentypen⁵ Zustand, Wächter und Zustandsgenerator generalisiert bezüglich den Datentypen der Umgebung und des Ereignisses.

Ereignis Ein Ereignis beschreibt den Zustand der Umgebung. Dieser ist durch den Spielzustand (gefallene Tore, Meldungen des Schiedsrichters) und den mechanischen Zustand (Objektpositionen und -geschwindigkeiten) beschrieben. Weiterhin trägt ein Ereignis Zusatzinformationen wie Position und Zeitstempel der letzten Erkennung eines verlorenen Objektes.

Zustand Ein Zustand stellt ein Teilverhalten von Spieler oder Mannschaft dar. Hierbei können aus einem Ereignis Steuerbefehle generiert und an den Empfänger gesendet werden. Der Rückgabewert der Ereignisverarbeitung durch einen Zustand zeigt an, ob das modellierte Teilverhalten erfolgreich abgeschlossen wurde, weiterhin abgearbeitet werden muss oder gescheitert ist.

Wächter Durch den Wächter werden opportunistische⁶ Verhaltensaspekte modelliert. Er prüft die Validität eines Zustandes im Bezug auf ein Ereignis. Hierdurch kann ein Zustand vorzeitig abgebrochen werden, um einen der aktuellen Situation besser angepassten Zustand zu aktivieren.

Zustandsgenerator Der Zustandsgenerator ist das Modell von Spieler- oder Mannschaftsverhalten. Wird ein Zustandsübergang durch erfolgreichen Abschluss oder Scheitern eines Teilverhaltens bzw. durch eine opportunistische Entscheidung signalisiert, dann erzeugt er aus diesem Signal ein Paar aus Folgezustand und Wächter für die aktuelle Situation. Der Zustandsgenerator implementiert somit einen endlichen Automaten.

Automat Der Automat kapselt die Ereignisverarbeitung durch Wächter und Zustand sowie den Ablauf eines Zustandsüberganges durch einen Zustandsgenerator. Ein Automat ist ein Zustand, wodurch sich eine kaskadierbare Struktur ergibt.

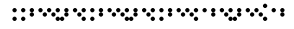
⁴ Abstrakte Datentypen zur Beschreibung von Schnittstellen nach [Str00]

⁵ Familien von Datentypen bezüglich Instanziierung durch Datentypen bzw. durch abzählbarer Mengen nach [Str00, cpp03]

⁶ Opportunismus wird in [BBBV05] als wichtige Eigenschaft des autonomen Ballsports technischer Systeme beschrieben.

1. Einleitung

Bezug zu objektorientierten Entwurfsmustern Aus der Sicht der Verwendung sind Zustand und Wächter Beobachter der Umgebung im Sinne des Beobachter-Entwurfsmusters [GHJV95]. Aus Sicht der Instanziierung sind Zustand und Wächter abstrakte Produkte. Ein Zustandsgenerator ist eine abstrakte Fabrik, er instanziiert Paare der abstrakten Produkte Zustand und Wächter [GHJV95]. Hierdurch kann der kapselnde Automat für beliebige Zustandsgeneratoren instanziiert werden, siehe Abbildung 1.6(b).



(G. C. Coleman)

Teil I.

Objektlokalisierung durch digitale Bildverarbeitung

2. Subsymbolische Verfahren der Bildverarbeitung

Die in der RoboCup Small Size League verwendeten Roboter verfügen über autonome Verfahren der Lagebestimmung, beispielsweise Odometrie über Inkrementalgeber an den Antrieben [Lan05, RF06, Jah09], optische Bodenabtastung [FSU11] oder Beschleunigungssensoren und Gyroskope [SPK⁺07]. Hiermit werden Geschwindigkeits- oder Lageregelkreise zur Beeinflussung der Roboterbewegung auf den Robotern selbst realisiert. Für den Spielbetrieb wird ein weiteres, absolutes Messsystem in Form eines digitalen Bildverarbeitungssystems eingesetzt. Dieses liefert mit Hilfe einer über dem Spielfeld angebrachten Kamera Informationen über die Zustände der Objekte auf dem Spielfeld und ermöglicht somit erst das koordinierte Spiel. Weiterhin werden durch das absolute Messsystem die lokalen Verfahren der Lagebestimmung im Spiel regelmäßig neu referenziert. Hierdurch reduziert sich die Problematik der Fehlerakkumulation der lokalen Lagebestimmungsverfahren erheblich.

Ein Zweig der Positions- oder Lagebestimmungsverfahren durch digitale Bildverarbeitung wird von modellbasierten statistischen Varianten der Bildauswertung gebildet. Vertreter hiervon sind Korrelationsanalysen, durchgeführt im Orts- oder Frequenzbereich, und die Hough-Transformation [BBD⁺09]. Beide Varianten finden ein Übereinstimmungsmaximum zwischen einem Modell der Umgebung und den Eingabedaten in einem diskreten Lösungsraum. Der Algorithmus RANSAC [FB81] ist verwandt zur Hough-Transformation bezüglich der Bestimmung eines Übereinstimmungsmaximums zwischen Modell und Eingabedaten, spart dabei aber Berechnungszeit dadurch, dass die untersuchten Lösungen aus den Eingabedaten selbst generiert werden. Während Hough-Transformation und RANSAC normalerweise auf zweiwertigen Bilddaten angewendet werden, erlauben Korrelationsanalysen auf mehrwertigen Bildern auch die Bestimmung von Texturinformationen. Während die statistischen Verfahren der Bildauswertung in der Lage sind, benötigte Informationen auch aus stark degenerierten Bildsignalen zu extrahieren, sind sie sehr rechenaufwendig.

Ein zweiter Zweig der Lagebestimmungsverfahren wird durch Segmentierungsmethoden aufgespannt, welche ein Eingabebild in Regionen gleicher Merkmalsausprägung zerlegen. Diese Regionen werden dann zur Lagebestimmung mit einem Modell abgeglichen. Segmentierungsmethoden liegen als ortsorientier-

2. Subsymbolische Verfahren der Bildverarbeitung

te [BBV00, Ler08], umrissorientierte [Ren09] Varianten vor oder finden im Frequenzbereich statt [LW95, MTJAL05]. Ein weiterer Zugang zur Bildsegmentierung kann in der Verwendung von Clusterverfahren [YGZ03, YSL09] gesehen werden, welche eng mit den ortsorientierten Verfahren verwandt sind.

Die Eignung eines Verfahrens zur Objektlokalisierung für den Einsatz im Roboter-Fußballspiel hängt entscheidend von der benötigten Rechenzeit ab. Um den Bedarf an Rechenzeit möglichst niedrig zu halten wird daher Abstand von den statistischen Auswertemethoden¹ genommen. Statt dessen wird aus dem Zweig der segmentorientierten Verfahren eine auf der Lauflängenkodierung eines klassifizierten Farbbildes basierende Variante nach [BBV00] gewählt, da sowohl die Segmentierungsaufgabe selbst, als auch die Implementierung vergleichsweise einfach und der erwartete Berechnungsaufwand niedrig ist.

Dieses Kapitel widmet sich dem subsymbolischen Teil der Bildauswertung, in dem der Übergang von digitalen Rohdaten der Kamera bis hin zu den Symbolen des Bildes, den Segmenten geleistet wird. Im Kapitel 3 werden diese Daten entsprechend eines Modells der Umgebung interpretiert und somit auf den Zustand der Umgebung geschlossen.

2.1. Vorverarbeitung

2.1.1. Rohdatenaufbereitung durch Bayerinterpolation

Zwei verbreitete Varianten digitaler Bildaufnahmesysteme sind 3-Chip- und 1-Chip-Kameras. Bei der 3-Chip-Kamera wird das Bild über Prismen bzw. Strahlteilungsspiegel auf drei separate Bildwandler projiziert, sodass je ein Sensor zur Detektion eines Farbkanals (Rot, Grün, Blau) genutzt wird, siehe Abbildung 2.1(a). Hierdurch steht für jedes Bildelement die vollständige Farbinformation zur Verfügung. Da hierfür ein exakt ausgerichtetes, aufwändiges Strahlteilungssystem sowie drei Bildwandler benötigt werden, sind 3-Chip-Kameras vergleichsweise teuer.

Eine Alternative hierzu stellen 1-Chip-Kameras mit Bayer-Sensor dar. Diese kommen mit einem einzigen Bildwandler und ohne Strahlteilungssystem aus. Nach einem bestimmten Muster (siehe Abbildung 2.1(b)) sind hier Farbfilter vor den einzelnen Bildelementen angeordnet.

¹In [BBD⁺09] wird die Einsatzfähigkeit statistischer Bildauswertungsverfahren unter Echtzeitanforderungen demonstriert. Diese werden dabei aber auf begrenzte Bildausschnitte unter Vorwissen über die zu erwartenden Objektlagen angewendet. Dieses Vorwissen kann entweder über einen vorangehenden Segmentierungsschritt mit entsprechend zusätzlichem Berechnungsaufwand oder durch eine Prediktion anhand der bisherigen Bewegung bereitgestellt werden. Im letzteren Fall ist es schwierig, Robustheit bei hohen Objektgeschwindigkeiten zu erreichen.

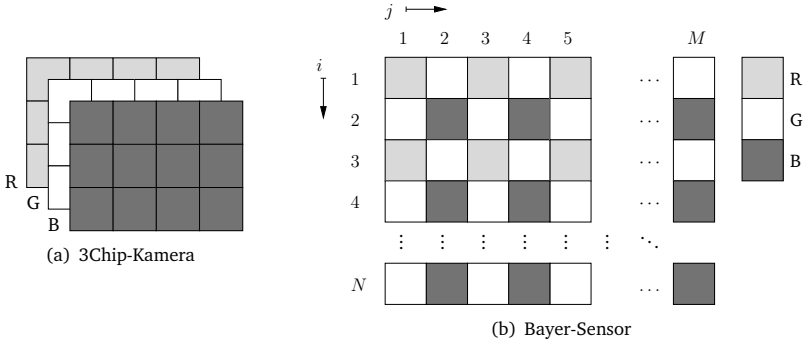


Abbildung 2.1.: Kameravarianten

Ein Bildelement weist abhängig von seinem Ort eine selektive Empfindlichkeit für Rot, Grün oder Blau auf, dies entspricht einem durch ein Farbfiler beeinflussten Maximum der spektralen Empfindlichkeit des Elementes im jeweiligen Farbbereich. Dem Vorteil der im Vergleich zu den 3-Chip-Kameras geringeren Fertigungskosten steht also der Nachteil gegenüber, dass die Farbinformationen unvollständig detektiert werden und somit eine Rekonstruktion der jeweils fehlenden Farbinformationen benötigt wird. Für die Rekonstruktion der fehlenden Farbinformationen existiert ein breites Spektrum ausgehend von sehr schnellen bis hin zu sehr genauen Verfahren, siehe [GGA⁺05] für eine instruktive Übersicht.

Das Bayermuster, als eine bestimmte Anordnung der Farbfiler

$F_{i,j} := F(i,j) \in \{R, G, B\}$ auf einer Matrix $V \in \mathbb{R}^{N \times M}$ der skalaren Bildelemente $v_{i,j} := v(i,j) \in \mathbb{R}$, kann zu

$$F_{i,j} = \begin{cases} B, & j \bmod 2 + i \bmod 2 = 0 \\ G, & j \bmod 2 + i \bmod 2 = 1 \\ R, & j \bmod 2 + i \bmod 2 = 2 \end{cases} \quad (2.1)$$

beschrieben werden, je nach verwendetem Sensor oder Bildausschnitt können Verschiebungen um 1 auftreten.

Wird mit $\vec{v} := (\mathbf{v}_R \ \mathbf{v}_G \ \mathbf{v}_B)^T \in \mathbb{R}^3$ der Wert eines zu rekonstruierenden Bildelementes der Komponenten $\mathbf{v}_R, \mathbf{v}_G, \mathbf{v}_B$ in den einzelnen Farbkäna len bezeichnet, dann muss ein Bayer-Interpolationsschema

2. Subsymbolische Verfahren der Bildverarbeitung

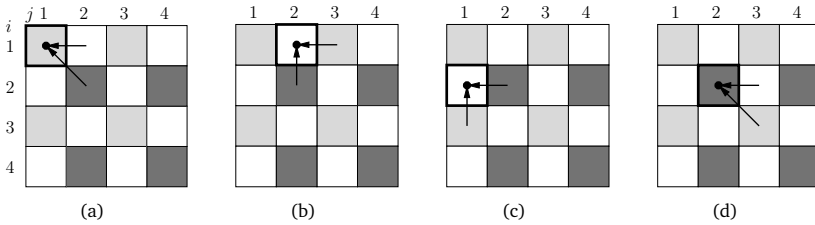


Abbildung 2.2.: Nächster-Nachbar-Interpolation

interpoliere : $\mathbb{R}^{N \times M} \times \mathbb{N}^2 \rightarrow \mathbb{R}^3 : (V, i, j) \mapsto \text{interpoliere}(V, i, j) = \vec{v}_{i,j}$ unter Berücksichtigung von (2.1) zur Verfügung stehen.

Die drei im folgenden vorgestellten einfachen Ortsbereichsverfahren zur Interpolation der fehlenden Farbinformationen zeichnen sich durch niedrigen Berechnungsaufwand aus.

Nächster-Nachbar-Interpolation

Bei der Nächster-Nachbar-Methode (siehe Abbildung 2.2 bzw. Gleichung (2.2)) werden die jeweils zwei fehlenden Farbkanäle von benachbarten Bildelementen direkt übernommen. Hierdurch stehen die drei Farbauszugsebenen mit achsweise halbiert Auflösung zur Verfügung. Die Ursprünge der drei Ebenen sind um jeweils ein Element in i - oder j -Richtung gegeneinander verschoben. Hierdurch entstehen Artefakte, die besonders an Kanten auffällig werden. Dem steht der Vorteil der niedrigen Bearbeitungszeit gegenüber.

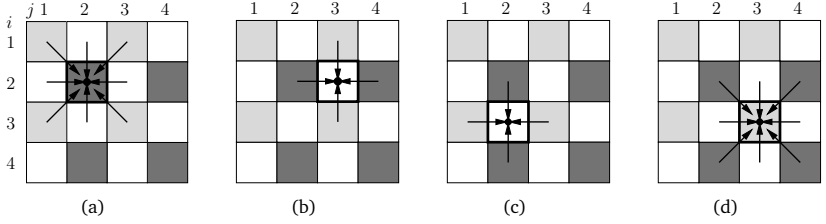


Abbildung 2.3.: Gemischt lineare/bilineare Interpolation

$$\begin{pmatrix} \mathbf{v}_R \\ \mathbf{v}_G \\ \mathbf{v}_B \end{pmatrix}_{i,j} = \begin{cases} \begin{pmatrix} v_{i+1,j+1} \\ v_{i,j+1} \\ v_{i,j} \end{pmatrix}, & (i \bmod 2 = 0) \wedge (j \bmod 2 = 0) \\ \begin{pmatrix} v_{i+1,j} \\ v_{i,j} \\ v_{i,j+1} \end{pmatrix}, & (i \bmod 2 = 0) \wedge (j \bmod 2 = 1) \\ \begin{pmatrix} v_{i,j+1} \\ v_{i,j} \\ v_{i+1,j} \end{pmatrix}, & (i \bmod 2 = 1) \wedge (j \bmod 2 = 0) \\ \begin{pmatrix} v_{i,j} \\ v_{i,j+1} \\ v_{i+1,j+1} \end{pmatrix}, & (i \bmod 2 = 1) \wedge (j \bmod 2 = 1) \end{cases} \quad (2.2)$$

Gemischt lineare/bilineare Interpolation

Das Schema der gemischt linearen/bilinearen Interpolation (Abbildung 2.3 bzw. Gleichung (2.3)) unterscheidet sich gegenüber der Nächster-Nachbar-Methode dadurch, dass die Werte der benachbarten Bildelemente nicht direkt übernommen werden, sondern dass durch lineare (für $i \bmod 2 + j \bmod 2 = 1$) bzw. bilineare (für $i \bmod 2 + j \bmod 2 \neq 1$) Interpolation eine Abschätzung der fehlenden Komponenten für den aktuell untersuchten Messort vorgenommen wird. Die Bildung von Artefakten an Kanten wird bereits hierdurch gegenüber der Nächster-Nachbar-Methode deutlich reduziert. Für die Berechnung der Farbwerte eines Bildelementes werden im Mittel $\frac{9+5}{2} = 7$ Werte benötigt. Der resultie-

2. Subsymbolische Verfahren der Bildverarbeitung

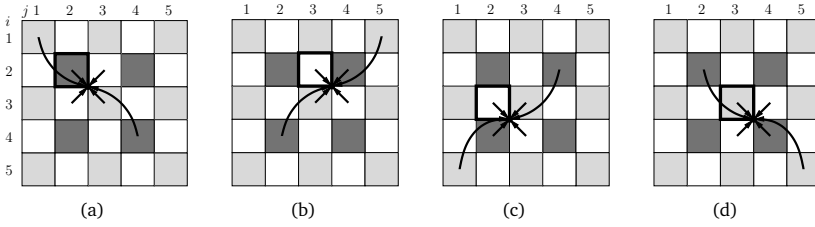


Abbildung 2.4.: Lineare Interpolation

rende mittlere Abstand² der zur Berechnung herangezogenen Bildelemente ist
 $E_{\text{linear/bilinear}} = \frac{28+10\sqrt{2}}{45} \approx 0,937$.

$$\begin{pmatrix} \mathbf{v}_R \\ \mathbf{v}_G \\ \mathbf{v}_B \end{pmatrix}_{i,j} = \frac{1}{4} \begin{cases} \begin{pmatrix} v_{i-1,j-1} + v_{i-1,j+1} + v_{i+1,j-1} + v_{i+1,j+1} \\ v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j} \\ 4v_{i,j} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 0) \\ \wedge (j \bmod 2 = 0) \end{matrix} \\ \begin{pmatrix} 2v_{i-1,j} + 2v_{i+1,j} \\ 4v_{i,j} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 0) \\ \wedge (j \bmod 2 = 1) \end{matrix} \\ \begin{pmatrix} 2v_{i,j-1} + 2v_{i,j+1} \\ 4v_{i,j} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 1) \\ \wedge (j \bmod 2 = 0) \end{matrix} \\ \begin{pmatrix} 4v_{i,j} \\ v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j} \\ v_{i-1,j-1} + v_{i-1,j+1} + v_{i+1,j-1} + v_{i+1,j+1} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 1) \\ \wedge (j \bmod 2 = 1) \end{matrix} \end{cases} \quad (2.3)$$

Lineare Interpolation

Eine rein lineare Interpolationsvariante wird in Abbildung 2.4 bzw. Gleichung (2.4) dargestellt. Während die gemischt lineare/bilineare Interpolation sich direkt auf die Bildelemente des Sensors bezieht, sind die Ergebnisse dieses Verfahrens auf die Zentren der Bayerzellen bezogen, also in i - und j -Richtung um jeweils ein halbes Bildelement verschoben. Zur Berechnung eines Bildelementes werden 6 Bildelemente benötigt, unter Einbeziehung der verwendeten

²Abbildung 2.3 verdeutlicht, dass im linearen Fall 5 Bildelemente mit mittlerem Abstand $E_{\text{linear}} = \frac{4}{5}$ und im bilinearen Fall 9 Bildelemente mit mittlerem Abstand $E_{\text{bilinear}} = \frac{4+4\sqrt{2}}{9}$ zur Berechnung herangezogen werden.

Wichtungen besteht ein mittlerer Abstand der verwendeten Bildelemente zur jeweiligen Bayerzelle von $E_{\text{linear}} = \frac{1}{6}(3\sqrt{\frac{1}{2}} + 2\sqrt{\frac{1}{2}} + 3\sqrt{\frac{1}{2}}) = \frac{4}{3}\sqrt{\frac{1}{2}} \approx 0,943$. Der mittlere Abstand der verwendeten Bildelemente zum Ort der Interpolation ist mit $E_{\text{linear}} = \frac{15}{7\sqrt{2}+5}E_{\text{linear/bilinear}} \approx 1,007E_{\text{linear/bilinear}}$ nur geringfügig größer als bei der gemischt linearen/bilinearen Interpolation. Dem gegenüber sinkt die Anzahl der zur Interpolation verwendeten Bildelemente auf $\frac{6}{7}$, was einen kleinen Vorteil in Bezug auf benötigte Speicherbandbreite darstellt.

$$\begin{pmatrix} \mathbf{v}_R \\ \mathbf{v}_G \\ \mathbf{v}_B \end{pmatrix}_{i,j} = \frac{1}{4} \begin{cases} \begin{pmatrix} 3v_{i+1,j+1} + v_{i-1,j-1} \\ 2v_{i+1,j} + 2v_{i,j+1} \\ 3v_{i,j} + v_{i+2,j+2} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 0) \\ \wedge (j \bmod 2 = 0) \end{matrix} \\ \begin{pmatrix} 3v_{i+1,j} + v_{i-1,j+2} \\ 2v_{i,j} + 2v_{i+1,j+1} \\ 3v_{i,j+1} + v_{i+2,j-1} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 0) \\ \wedge (j \bmod 2 = 1) \end{matrix} \\ \begin{pmatrix} 3v_{i,j+1} + v_{i+2,j-1} \\ 2v_{i,j} + 2v_{i+1,j+1} \\ 3v_{i+1,j} + v_{i-1,j+2} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 1) \\ \wedge (j \bmod 2 = 0) \end{matrix} \\ \begin{pmatrix} 3v_{i,j} + v_{i+2,j+2} \\ 2v_{i+1,j} + 2v_{i,j+1} \\ 3v_{i+1,j+1} + v_{i-1,j-1} \end{pmatrix}, & \begin{matrix} (i \bmod 2 = 1) \\ \wedge (j \bmod 2 = 1) \end{matrix} \end{cases} \quad (2.4)$$

2.1.2. Helligkeitsausgleich

Bedingt durch ungleichmäßige Ausleuchtung des Spielfeldes und Randabschattungen des Objektivs erfolgt ein systematischer örtlicher Helligkeitsverlauf. Hierzu überlagert sind durch reale elektronische Bauteile verursachte zufällige Effekte. Bei isolierter Betrachtung eines Bildelementes ist es nicht möglich, dessen Farbwert immer der richtigen Farbklassen zuzuordnen, wenn dem Betrachter keine Zusatzinformationen (Nachbarfarbwerte, Kantenverläufe, Position auf dem Feld) zur Verfügung stehen. Da eine Hauptanforderung an das Bildverarbeitungssystem die möglichst hohe Verarbeitungsgeschwindigkeit darstellt, soll auf die Nutzung von Nachbarschaftsbeziehungen, wie sie beispielsweise durch Operationen wie Kantenfindung oder lokale Mittelwerte modelliert werden können, möglichst verzichtet werden. Zur Kompensation der systematischen lokalen Effekte bietet sich einerseits eine ortsabhängige Klassifikation und andererseits ein lokaler Helligkeitsausgleich mit anschließender ortsunabhängiger Klassifikation an. Erstere Variante ist erfolgversprechend, wenn die systematischen Effekte nicht nur die Bildhelligkeit betreffen, also beispielsweise auch schlecht

2. Subsymbolische Verfahren der Bildverarbeitung

korrigierbare ortsabhängige Farbverschiebungen vorliegen. Im Gegenzug steigt der Trainings- und Lernaufwand für die Bestimmung der Klassifikatoren. Da am vorliegenden Versuchsstand das größte Problem in der ortsabhängigen Bildhelligkeit liegt, wird statt dessen die zweite Variante, der statische lokale Ausgleich, umgesetzt.

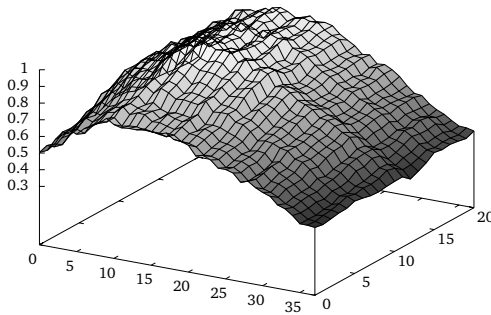


Abbildung 2.5.: Helligkeitsverlauf am verwendeten Spielfeld: Dem Bildausschnitt des Spielfeldes ist ein Gitter mit 37×20 Elementen überlagert. Die dargestellte Helligkeit ist der Mittelwert aller zu einer Zelle gehörenden Bildelemente, normiert auf die Zelle des größten Mittelwertes.

Statischer lokaler Ausgleich

Als Varianten des lokalen Helligkeitsausgleiches bieten sich die Differenzbildmethode und die Verwendung ortsabhängiger Verstärkungen an. Die Idee der Differenzbildmethode ist es, in der Vorbereitungsphase ein Referenzbild der Umgebung ohne die zu detektierenden Objekte aufzunehmen. Dieses Bild kodiert die systematischen lokalen Effekte. In der Nutzphase wird dieses Referenzbild von jedem Bild des Videostroms elementweise subtrahiert. Der Helligkeitsausgleich ist dabei eher ein Nebeneffekt des Verfahrens, welches Objekte gut vom Hintergrund trennt. Ein Problem für die Farbbildverarbeitung kann sich dabei in einer ortsabhängigen Drift der Farbwinkel manifestieren.

Bei der Alternative, der Verwendung einer ortsabhängigen Verstärkung werden die Kehrwerte der Matrix der mittleren Helligkeiten (siehe Abbildung 2.5)

als Verstärkungsfaktoren für die jeweiligen Bildteile benutzt, so ergibt sich ein statischer lokaler Helligkeitsausgleich ohne Farbverschiebungen. Dieses Verfahren hat sich im praktischen Einsatz als geeignet erwiesen.

Dynamische globale Verstärkung

Zusätzlich zum Ausgleich lokaler Helligkeitsunterschiede im Bild soll das Bildverarbeitungssystem in der Lage sein, globale Schwankungen der Bildhelligkeit auszugleichen. Es wird versucht, die tatsächliche mittlere Bildhelligkeit I den Trainingsbedingungen anzunähern.

Die verwendete Kamera verfügt über zwei Stellmöglichkeiten zur Helligkeitssteuerung: Die Integrationszeit der Kamera τ und eine Signalverstärkung $g \in [0, 30] \subset \mathbb{N}$, welche in ganzzahligen dB eingestellt werden kann. Die minimal zulässige Bildwiederholfrequenz bzw. die tolerierbare Bewegungsunschärfe bei Maximalgeschwindigkeit bestimmt die maximale zulässige Integrationszeit τ_{\max} . Aus den aktuellen Kameraparametern g und τ folgt die aktuelle verstärkungsbereinigte Ersatzintegrationszeit τ_e zu

$$\tau_e = \tau 10^{g/20}. \quad (2.5)$$

Das Verhältnis geforderter mittlerer Bildhelligkeit I_r und vorliegender Bildhelligkeit als arithmetischem Mittel der Bildelemente führt zur geforderten verstärkungsbereinigten Ersatzintegrationszeit

$$\tau'_r = \frac{\tau_e I_r}{\frac{1}{NM} \sum_{v_{i,j} \in V} [u_{i,j}]}. \quad (2.6)$$

Die Signalübertragung zwischen Kamera und Bildverarbeitungsrechner ist mit einer Totzeit behaftet. Werden die geforderten Belichtungsparameter ungedämpft eingestellt, neigt das System zum Schwingen. Durch Einsatz eines experimentell bestimmten Dämpfungsfaktors $0 < k_\tau < 1$ kann die einzustellende Ersatzintegrationszeit τ_r zu

$$\tau_r = (1 - k_\tau) \tau'_r + k_\tau \tau_e \quad (2.7)$$

bestimmt werden. Hiermit ergibt sich die erwünschte Verstärkung g'' zu

$$g'' = 20 \log_{10} \left(\frac{\tau_r}{\tau_{\max}} \right). \quad (2.8)$$

2. Subsymbolische Verfahren der Bildverarbeitung

Die erwünschte Verstärkung g'' ist im allgemeinen keine Natürliche Zahl. Deshalb wird die realisierbare Verstärkung g' zu

$$g' = \begin{cases} 0, & g'' < 0 \\ g_{\max}, & g'' > g_{\max} \\ \lceil g'' \rceil, & \text{sonst} \end{cases} \quad (2.9)$$

bestimmt. Hiermit wird es möglich, die zu stellende Integrationszeit τ' zu

$$\tau' = \tau_{\max} 10^{(g'' - g')/20} \quad (2.10)$$

zu bestimmen. Die Belichtungsparameter g' und τ' werden nun an die Kamera übertragen.

2.2. Farbklassifikation

Das Ziel der Bildauswertung durch digitale Bildverarbeitung ist die Lokalisierung und Identifizierung der Objekte der Spielumgebung. Hierzu wird die Lage vordefinierter Markierungen im Bild ausgewertet. Die Markierungen zeichnen sich durch die Ausprägung des Merkmals *Farbe* aus.

Die Farbe des Balles ist dabei Orange, zur Kodierung der Mannschaftszugehörigkeit eines Spielers sind die Farben Gelb und Blau vorgesehen. Mit den Farben Grün, Cyan, Magenta und Weiß dürfen Spielernummer und Ausrichtung kodiert werden. Die Menge der zu unterscheidenden Farben ist somit $\mathfrak{C} = \{\text{Weiß, Orange, Gelb, Grün, Cyan, Blau, Magenta, Rest}\}$, wobei die Elemente dieser Menge einen Konsens menschlicher Wahrnehmung darstellen. Die Klasse Rest ist reserviert für alle nicht zuordenbaren Farbproben. Die zu unterscheidenden Markierungsfarben $\mathfrak{C} \setminus \text{Rest}$ sollen eine gute Detektierbarkeit der im Spiel vorkommenden Farben durch RGB-Kameras gewähren, da alle bunten Farben des Spieles im wesentlichen den Primärvalenzen der Kameras oder deren Kom-

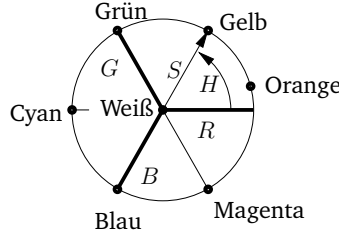


Abbildung 2.6.: Idealierte Darstellung der markierenden Farbklassen im Farbkreis mit Tonwinkel H (Hue) und Sättigung S (Saturation).

plementen direkt entsprechen. Abbildung 2.6 zeigt hierzu eine schematische Darstellung der Markierungsfarben in der H-S-Ebene des HSV-Farbraums³.

Im Schritt der Bayerinterpolation wurde eine Rekonstruktion der RGB-Werte $\vec{v}_{i,j}$ für jedes Bildelement $v_{i,j} \in V$ vorgenommen. Hiermit steht eine Näherung des tatsächlichen Bildinhaltes zur Verfügung. Bei der folgenden segmentbasierten Bildauswertung sollen die Spielermarkierungen als zusammenhängende Gebiete der Markierungsfarben im Videostrom identifiziert werden. Für eine effiziente Bildauswertung ist es zweckmäßig, diese Segmentierung anhand der Zugehörigkeit der Bildelemente $\vec{v}_{i,j}$ zu den vorkommenden Farbklassen vorzunehmen.

Durch Fertigungstoleranzen, Inhomogenität der Ausleuchtung, Abbildungsfehler, Sensor- und Verstärkungsrasschen, Quantisierungs- und Interpolationsfehler werden die einzelnen Farben nicht scharf in den Farbraum des Bildverarbeitungssystems abgebildet. Stattdessen werden verschiedene Bereiche des Farbraumes von Vertretern der jeweiligen Farbklassen dominiert.

Die Zuordnung eines Bildelementes $\vec{v}_{i,j}$ zu der entsprechenden Farbklasse $C \in \mathcal{C}$ ist die Klassifizierungsaufgabe. Sie wird in vorliegender Arbeit durch

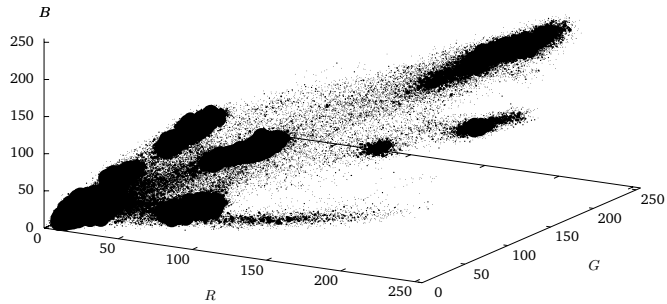
³Die Transformation aus dem RGB- in den HSV-Farbraum nach [FDSH97] ermöglicht die Auswertung der Komponenten *Hue*-Tonwinkel, *Saturation*-Sättigung und *Value*-Helligkeit. Hierfür seien $M := \max(\mathbf{v}_R, \mathbf{v}_G, \mathbf{v}_B)$ und $D := M - \min(\mathbf{v}_R, \mathbf{v}_G, \mathbf{v}_B)$.

$$\text{hsv}(\vec{v}) := \begin{pmatrix} \frac{\pi}{3} \begin{cases} 0 + (\mathbf{v}_G - \mathbf{v}_B)/D, & (M = \mathbf{v}_R) \wedge (D > 0) \\ 2 + (\mathbf{v}_B - \mathbf{v}_R)/D, & (M = \mathbf{v}_G) \wedge (D > 0) \\ 4 + (\mathbf{v}_R - \mathbf{v}_G)/D, & (M = \mathbf{v}_B) \wedge (D > 0) \\ \text{nicht definiert,} & D = 0 \end{cases} \\ \begin{cases} D/M, & M > 0 \\ \text{nicht definiert,} & M = 0 \end{cases} \\ M \end{pmatrix} \quad (2.11)$$

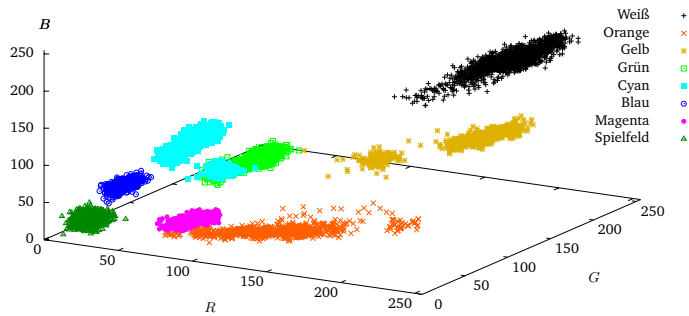
2. Subsymbolische Verfahren der Bildverarbeitung

Zerlegung des Farbraumes anhand zugeordneter Stichproben gelöst. Diese Realisierung fällt somit in den Bereich des *überwachten Lernens*. Abbildung 2.7 zeigt eine Gegenüberstellung der Streudiagramme der Farbwerte des unklassifizierten Bildes bzw. der nutzerklassifizierten Stichproben. Mit den in Abschnitt 2.2.2 vorgestellten Verfahren wird versucht, eine autonome Stichprobenzuordnung zu realisieren. Dieser Übergang in die Domäne des *unüberwachten Lernens* liefert mit den vorgestellten Verfahren jedoch keine für einen robusten Spieleinsatz ausreichenden Ergebnisse, deshalb erfolgt die Stichprobenzuordnung weiterhin durch einen menschlichen Lehrer.

2.2. Farbklassifikation



(a)



(b)

Abbildung 2.7.: Streudiagramme der auftretenden Farbwerte im RGB-Farbraum: unklassifiziert (a) und nutzerklassifiziert (b).

2. Subsymbolische Verfahren der Bildverarbeitung

2.2.1. Klassifikation durch Generalisierung von Stichproben

Einhüllende Quader

Eine Möglichkeit zum Aufbau eines Klassifikators aus Stichproben besteht in der komponentenweisen Bestimmung einer unteren und oberen Schranke für die Zugehörigkeit zu einer Farbkasse [BBV00, MDSG02], siehe Abbildung 2.8. Die Klassifikation muss dabei nicht im RGB-Farbraum erfolgen. In [BBV00] wird sie erfolgreich im YUV-Farbraum eingesetzt.

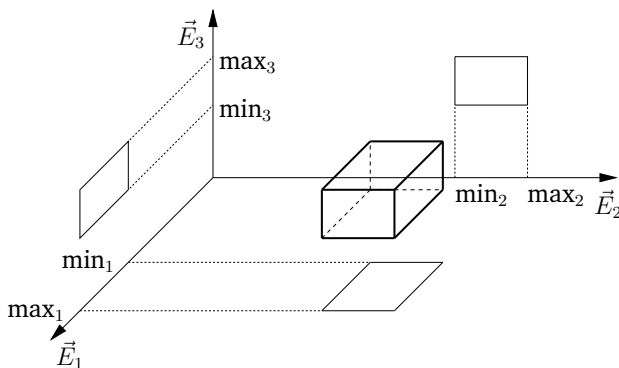


Abbildung 2.8.: Minimale einhüllende Quader

Als Schwellwerte bieten sich die durch die Stichprobe im betrachteten Farbraum realisierten Extremalwerte an, sie können aber auch aus komponentenweisen statistischen Momenten der Stichproben gebildet werden. Dieses leicht implementierbare und effiziente Verfahren ist gut anwendbar, wenn sich die resultierenden Quader nicht überlappen.

Die hier verwendete Kamera stellt die Bilddaten im RGB-Farbraum zur Verfügung. In diesem stellen sich die Verteilungen der zu unterscheidenden Farbklassen als Linien oder lang gezogene Ellipsoide dar, welche sich nicht überlappungsfrei durch einhüllende Quader trennen lassen. Zum Vermeiden von Überlappungen kann einerseits die Abdeckung der Verteilungen durch die Quader reduziert werden, was zu fehlerhaften Zuordnungen zur Klasse Rest führt. Andererseits kann die Klassifikation in einem besser geeigneten Farbraum durchgeführt werden – dann entsteht entweder zusätzlicher Berechnungsaufwand für die Transformation in den Zielfarbraum oder zusätzlicher Bedarf an Speicherbandbreite falls die Klassifikationsergebnisse bereits als vorberechnete Tabelle vorliegen. Beide Auswege führen zum Verlust der Vorteile des Verfahrens.

Diskrete Voronoidekomposition

Die Voronoidekomposition ist eine Zerlegung einer Menge in disjunkte Teilmengen. Ist Q die bezüglich der Menge $\Xi = \{\xi_1, \dots, \xi_n\}$ und der Distanz $d : Q \times \Xi \mapsto \mathbb{R}$ in disjunkte Voronoizellen $\{\mathcal{V}_1, \dots, \mathcal{V}_n\}$ zu partitionierende Menge, dann liefert die Voronoidekomposition die Zerlegung von Q , in der alle $q \in \mathcal{V}_i$ einen kleineren Abstand $d(q, \xi_i)$ zu ξ_i haben, als zu allen Elementen von $\Xi \setminus \xi_i$ (vergleiche [Sud06]):

$$\mathcal{V}_i = \{q \in Q \mid d(q, \xi_i) < d(q, \xi_j), j \neq i\}. \quad (2.12)$$

Die Zerlegung nach (2.12) erfolgt nach [Sud06], im Unterschied zur dortigen Formulierung gehört das aus den Zerlegungsgrenzen bestehende Skelett hier nicht zu den Voronoizellen – die Voronoizellen sind also offene Mengen.

Bei der Menge Ξ kann es sich um eine Menge von punktförmigen „Wachstumskeimen“, um explizit beschriebene Verteilungen, aber auch um Stichproben der Farbklassen handeln.

Allgemein erfolgt die Generalisierung einer Stichprobe $S_C = (\vec{c}_1, \dots, \vec{c}_{k_C})$, $\vec{c}_i \in \mathbb{R}^n$ durch Voronoidekomposition zu

$$\begin{aligned} &\text{Generalisierung} : \mathbb{R}^n \rightarrow \mathfrak{C} \\ \vec{v} &\mapsto \text{Generalisierung}(\vec{v}) = \arg \min_{C \in \mathfrak{C}} (d(\vec{v}, S_C)). \end{aligned} \quad (2.13)$$

Die partitionierende Menge Ξ entspricht der Menge der Stichproben $\{S_i\}$ der Farbklassen und die partitionierte Menge Q einem Farbraum.

Bayes-Klassifikation Die Realisierung eines Messwertes \vec{v} kann als Zufallsexperiment aufgefasst werden, bei dem dieser Wert verdeckt aus der Menge der Klassen \mathfrak{C} gezogen wird. Zugeordnete Trainingsdaten S_C für eine Klasse C repräsentieren eine Verteilung $P(\vec{v}|C)$, die bedingte Wahrscheinlichkeit dafür, dass \vec{v} realisiert wird, wenn C die generierende Klasse ist. Aus den geometrischen Rahmenbedingungen (Spielfeldgröße, zu erwartende Größen der jeweiligen Markierungen) können die Wahrscheinlichkeiten $P(C)$, dafür dass ein Messwert durch Klasse C generiert wird, zu $P(C) \approx A_C / \sum_{D \in \mathfrak{C}} [A_D]$ geschätzt werden. Für einen gegebenen Wert \vec{v} wird mit Hilfe des Satzes von Bayes bestimmt, mit welcher Wahrscheinlichkeit $P(C|\vec{v})$ dieser Wert durch die Klasse C realisiert wurde:

$$P(C|\vec{v}) = \frac{P(C \cap \vec{v})}{P(\vec{v})} = \frac{\frac{P(C \cap \vec{v})}{P(C)} P(C)}{P(\vec{v})} = \frac{P(\vec{v}|C) P(C)}{P(\vec{v})}, \quad (2.14)$$

2. Subsymbolische Verfahren der Bildverarbeitung

wobei die totale Wahrscheinlichkeit $P(\vec{v})$ des Auftretens von \vec{v} über alle Klassen hinweg bestimmt ist durch

$$P(\vec{v}) = \sum_{C \in \mathfrak{C}} [P(\vec{v}|C)P(C)]. \quad (2.15)$$

Das Risiko $r_C(\vec{v})$ einer Fehlzuordnung ist

$$r_C(\vec{v}) = \sum_{D \in \mathfrak{C}} [L_{C,D}P(D|\vec{v})], \quad (2.16)$$

wobei $L_{C,D}$ die Kosten einer Fehlzuordnung zwischen C und D beschreibt.

Werden die Fehlzuordnungskosten zu $L_{C,D} := \begin{cases} 1, & C \neq D \\ 0, & C = D \end{cases}$ gewählt, dann geht das Risiko $r_C(\vec{v})$ einer Fehlzuordnung in die Wahrscheinlichkeit $\bar{P}(C|\vec{v})$ einer Fehlzuordnung über:

$$r_C(\vec{v}) = \sum_{D \in \mathfrak{C}, C \neq D} [P(D|\vec{v})] = \bar{P}(C|\vec{v}) = 1 - P(C|\vec{v}). \quad (2.17)$$

Das minimale Fehlklassifikationsrisiko wird dann bei maximaler a-posteriori-Wahrscheinlichkeit erzielt

$$\arg \min_{C \in \mathfrak{C}} (r_C(\vec{v})) = \arg \max_{C \in \mathfrak{C}} (P(C|\vec{v})), \quad (2.18)$$

weshalb die Wahrscheinlichkeit einer Fehlzuordnung $\bar{P}(C|\vec{v})$ als Distanz $d(\vec{v}, S_C)$ einer Probe \vec{v} zur Stichprobe S_C für die diskrete Voronoidekomposition in (2.13) benutzt wird:

$$d_B(\vec{v}, S_C) := \bar{P}(C|\vec{v}) = 1 - P(C|\vec{v}). \quad (2.19)$$

Hierdurch wird der Farbraum so durch die Verteilungen der Stichproben in Farbklassen partitioniert, dass das Risiko einer Fehlklassifikation für jedes Element \vec{v} minimal ist. Damit realisiert diese Dekompositionsvariante eine Bayes-Klassifikation, vergleiche [Zha00] für Gleichungen (2.15)-(2.18).

Für die Implementierung des Klassifikators muss eine Abschätzung von $S_C \mapsto P(C|\vec{v})$ gemacht werden. Für den praktischen Einsatz wird hierfür eine multivariate Normalverteilung zu

$$P(\vec{v}|C) = \frac{e^{-\frac{1}{2}(\vec{v}-\vec{\mu}_C)^T \Sigma_C^{-1}(\vec{v}-\vec{\mu}_C)}}{(2\pi)^{n/2} \sqrt{|\Sigma_C|}} \quad (2.20)$$

angesetzt, wobei die Parameter Klassenzentrum $\vec{\mu}_C$ und -kovarianzmatrix Σ_C zu

$$\vec{\mu}_C \approx \frac{1}{k_C} \sum_{i=1}^{k_C} [\vec{c}_i] \quad (2.21)$$

und

$$\Sigma_C \approx \frac{1}{k_C - 1} \sum_{i=1}^{k_C} \left[\begin{pmatrix} \Delta \mathbf{c}_1 \Delta \mathbf{c}_1 & \dots & \Delta \mathbf{c}_1 \Delta \mathbf{c}_n \\ \vdots & \ddots & \vdots \\ \Delta \mathbf{c}_n \Delta \mathbf{c}_1 & \dots & \Delta \mathbf{c}_n \Delta \mathbf{c}_n \end{pmatrix} \right], \quad \Delta \vec{c} := \vec{c}_i - \vec{\mu}_C \quad (2.22)$$

geschätzt werden. Die Anwendung von (2.14) auf die Annahmen normalverteilter Klassen nach (2.20) sowie der Auftretswahrscheinlichkeiten der Farbklassen entsprechend den Verhältnissen der Flächeninhalte liefert dann für eine Menge zugeordneter Stichproben $\{S_{\mathfrak{C}}\}$ den dekomponierenden Abstand (2.19).

Mahalanobisdistanz Aus den Gleichungen (2.18), (2.14) und (2.20) folgt für die Klassifikation des kleinsten Fehlzuordnungsrisikos

$$\begin{aligned} \arg \min_{C \in \mathfrak{C}} [r_C(\vec{\mathbf{v}})] &= \arg \max_{C \in \mathfrak{C}} [P(C|\vec{\mathbf{v}})] = \arg \max_{C \in \mathfrak{C}} [P(\vec{\mathbf{v}}|C)P(C)] \\ &= \arg \max_{C \in \mathfrak{C}} \left[|\Sigma_C|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{\mathbf{v}} - \vec{\mu}_C)^T \Sigma_C^{-1} (\vec{\mathbf{v}} - \vec{\mu}_C)} P(C) \right] \\ &= \arg \min_{C \in \mathfrak{C}} \left[\ln(|\Sigma_C|) + \underbrace{(\vec{\mathbf{v}} - \vec{\mu}_C)^T \Sigma_C^{-1} (\vec{\mathbf{v}} - \vec{\mu}_C)}_{=d_M^2(\vec{\mathbf{v}}, S_C)} - 2 \ln(P(C)) \right]. \end{aligned} \quad (2.23)$$

Unter der Annahme, dass $\ln(|\Sigma_C|) - 2 \ln(P(C)) \approx \text{konstant} \quad \forall C \in \mathfrak{C}$ gilt, vereinfacht sich (2.23) zu

$$\arg \min_{C \in \mathfrak{C}} [r_C(\vec{\mathbf{v}})] = \arg \min_{C \in \mathfrak{C}} d_M^2(\vec{\mathbf{v}}, S_C) = \arg \min_{C \in \mathfrak{C}} d_M(\vec{\mathbf{v}}, S_C), \quad (2.24)$$

wobei

$$d_M(\vec{\mathbf{v}}, S_C) := \sqrt{(\vec{\mathbf{v}} - \vec{\mu}_C)^T \Sigma_C^{-1} (\vec{\mathbf{v}} - \vec{\mu}_C)} \quad (2.25)$$

die Mahalanobisdistanz [Mah36] ist.

2. Subsymbolische Verfahren der Bildverarbeitung

Unter den getroffenen Annahmen stellt somit der Aufbau des Klassifikators (2.13) bereits durch die Mahalanobisdistanz eine gute Näherung eines Bayes-Klassifikators dar, solange

$$d_M^2(\vec{v}, S_D) - d_M^2(\vec{v}, S_C) > \ln \left(\frac{|\Sigma_C|}{|\Sigma_D|} \right) + 2 \ln \left(\frac{P(D)}{P(C)} \right) =: \Delta^2 \min_{C,D} \quad (2.26)$$

gilt, ist Separation zwischen C und D bayes-exakt.

Die Mahalanobisdistanz ermöglicht durch die vereinfachende Annahme eine plausible Klassifikation [ITM01, KO05] bei reduziertem benötigtem Vorwissen gegenüber der Bayes-Klassifikation. Eine kleine Distanz einer Probe \vec{v} von einer Klasse C entspricht einer kleinen Irrtumswahrscheinlichkeit für die Annahme der Realisierung von \vec{v} durch C .

Die Mahalanobisdistanz $d_M(\vec{v}, C)$ einer Klasse C hat die Struktur einer quadratischen Metrik, ihre Isoklinen bilden also Ellipsoide um $\vec{\mu}_C$. Diese Variante der Klassifikation wird am RoboCup-Versuchsstand der TU Ilmenau seit [Ler08] erfolgreich eingesetzt. Für den praktischen Einsatz wird dabei die Distanz (2.25) zur Klasse Rest mit einem experimentell zu bestimmenden Maximalabstand d_{\max} derart modifiziert, dass $d_M(\vec{v}, \text{Rest}) \leq d_{\max}$. Eine Probe, deren Minimalabstand zu allen anderen Klassen $\mathcal{C} \setminus \text{Rest}$ diesen Schwellwert d_{\max} übersteigt, wird somit der Klasse Rest zugeordnet.

Minimaler Stichprobenabstand Abbildung 2.9(a) zeigt ein synthetisches Beispiel zweier Verteilungen, von denen eine bimodal ist. Die Bayes-Klassifikation für Normalverteilungen sowie die von ihr abgeleitete Klassifikation anhand der Mahalanobisdistanz versagen hier. Einige Vertreter der unimodalen Verteilung haben einen größeren Mahalanobisabstand zu ihrer eigenen Verteilung als zur bimodalen, gleiches ist für den Bayes-Ansatz unter der Annahme einer Normalverteilung zu erwarten. Dieses Beispiel motiviert, die Abstandsdefinition nicht auf eine aus der Stichprobe generierte Beschreibung der Verteilung, sondern auf die Stichprobe selbst zu stützen.

Eine mögliche Realisierung ist der elementbezogene Minimalabstand eines Farbwertes zur Stichprobe:

$$d_S(\vec{v}, S_C) := \min_{i=1}^{k_C} [\mathfrak{d}(\vec{v}, \vec{c}_i)], \quad (2.27)$$

wobei $\mathfrak{d} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ eine geeignete Metrik, beispielsweise Manhattan-, euklidische oder Maximum-Metrik ist.

Wird d_S aus in (2.12) verwendet so ergibt sich die Voronoidekomposition über dem elementbezogenen Stichprobenabstand bzw. Nächster-Nachbar-Abstand.

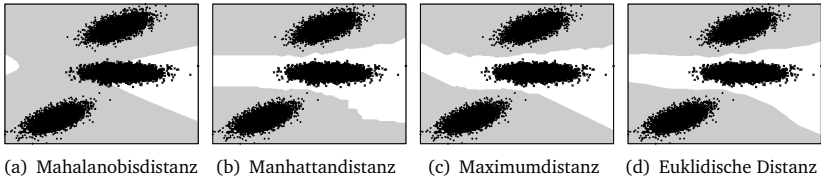


Abbildung 2.9.: Diskrete Voronoidekomposition anhand verschiedener Metriken, grau: Klasse 1 (bimodal), weiß: Klasse 2

Abbildung 2.9 zeigt dies am eingangs erwähnten Beispiel. Für Manhattan- und Maximummetrik ergeben sich hierbei Ergebnisse entsprechend einer morphologischen Dilatation⁴ der Stichprobe im Farbraum [Bau08], siehe Abbildungen 2.9(b) und 2.9(c).

Dieser Klassifikationsansatz wird in Betracht gezogen, da die Klassifikation anhand der Mahalanobisdistanz ohne Einsatz eines lokalen Helligkeitsausgleiches nicht geeignet ist, alle vorkommenden Farbklassen robust zu trennen. Beim stichprobenbasierten Minimalabstand entfällt die Annahme multivariater Normalverteilungen der Klassen, weshalb dieser Ansatz erfolgversprechend erscheint. [CH67] zeigt, dass der Erwartungswert der Fehlklassifikationen bei der Nächster-Nachbar-Klassifikation in den Grenzen zwischen einfachem und doppeltem Erwartungswert der Fehlklassifikationen beim Bayesklassifikator mit vollständigem Vorwissen über Struktur und Parameter der entsprechenden Verteilungen liegt. Dabei ist das einzige benötigte Vorwissen eine geeignete Metrik.

In der Praxis zeigt sich jedoch, dass auch dieser Ansatz ohne lokalen Helligkeitsausgleich nicht für die robuste Separation aller vorkommenden Klassen geeignet ist. Eine mögliche Ursache hierfür ist, dass die Klassifikation durch einen Nachbarn entschieden wird. Ausreißer bzw. Fehlzuordnungen in Trainingsdatensatz haben damit relativ starken Einfluss auf das Klassifikationsergebnis. Eine Verallgemeinerung des Nächster-Nachbar-Ansatzes zum k -Nächster-Nachbar-Ansatzes, bei dem die Klassifikation nicht am nächsten Nachbarn sondern an der Menge der k nächsten Nachbarn entschieden wird (vergleiche [DH04b]), kann diese Anfälligkeit mildern, wurde hier jedoch nicht untersucht. Statt dessen wird der Ansatz des lokalen Helligkeitsausgleiches gewählt, welcher das Klassifikationsproblem stark vereinfacht und sowohl mit Bayes-Ansatz, Dekomposition

⁴In [VMM09] wird die diskrete Voronoidekomposition als Ergebnis eines Dilatationsprozesses beschrieben.

2. Subsymbolische Verfahren der Bildverarbeitung



Abbildung 2.10.: Stichprobenzuordnung: Bisheriges manuelles Markieren und Zuordnen durch den Nutzer (a) sowie angestrebte automatische Zuordnung in einem festgelegten Bildausschnitt (b).

über der Mahalanobisdistanz, als auch mit der 1-Nächster-Nachbar-Methode praxistaugliche Ergebnisse erzielt.

2.2.2. Autonome Stichprobenerhebung

Die Eingabedaten für den Aufbau eines Klassifikators in Abschnitt 2.2.1 sind Stichproben als Vertreter der Verteilungen der zu unterscheidenden Farbklassen. Diese werden bisher durch manuelles Markieren und Zuordnen entsprechender Bereiche im Videobild gewonnen, siehe Abbildungen 2.7(b) und 2.10. Bisherige Erfahrungen haben gezeigt, dass die Qualität der resultierenden Klassifikatoren durch den menschlichen Trainer beeinflusst wird. Deshalb besteht der Wunsch, den Prozess der Stichprobenerhebung und Zuordnung zu automatisieren. Hierin kann ein Übergang zu Verfahren des unüberwachten Lernens gesehen werden. Eine Klasse hierfür erfolgversprechender Lösungen wird durch die Kohonenkarten [RMS91] bereitgestellt, dies sind neurobiologisch inspirierte Klassifikatoren, welche selbstständig eine Abbildung von einem hochdimensionalen Definitionsbereich in einen niederdimensionalen Wertebereich lernen. Die hierfür verwendeten Lernregeln sollen dabei sicherstellen, dass lokale Nachbarschaften der Elemente des Definitionsbereiches im Wertebereich erhalten bleiben.

Aufgrund der bisherigen guten Erfahrungen mit den in Abschnitt 2.2.1 dargestellten Klassifikatoren wird hier jedoch der Ansatz verfolgt, in einem ersten Schritt Partitionen der im Farbraum realisierten Elemente entsprechend der generierten Farben zu bestimmen und diese in einem zweiten Schritt den jeweiligen Farbklassen zuzuordnen. Somit stünden durch diese autonom erstellten und

zugeordneten Partitionen Stichproben der zu unterscheidenden Farbklassen für einen anschließenden Klassifikatoraufbau nach 2.2.1 bereit.

Für den Schritt der autonomen Stichprobenerhebung bzw. Stichprobenzuordnung werden im Folgenden Verfahren der Clusteranalyse bzw. Mustererkennung verwendet. Da bisher die den Farbklassen entsprechenden Stichproben mit keinem der folgend beschriebenen Verfahren ausreichend robust extrahiert werden können, wird der Schritt der Zuordnung der Partitionen zu den Farbklassen vorläufig nicht behandelt.

RANSAC

Random Sample Consensus [FB81] ist ein Verfahren, um in einer Menge von Eingabedaten Muster einer bestimmten Struktur, beispielsweise Geraden, Flächen oder Kreise, zu finden. Die Menge der Eingabedaten sei $D = \{p_i\} \subset \mathcal{D}$. Der Abstand eines Musters $\mathcal{M}_i \in \mathcal{S}$ der Struktur \mathcal{S} kann durch $d_R : \mathcal{D} \times \mathcal{S} \mapsto \mathbb{R}$ bestimmt werden. Die Menge der Unterstützer P_i^* des Musters \mathcal{M}_i bezüglich eines Maximalabstandes ε ergibt sich somit zu $P_i^* = \{p_j \in D \mid d_R(p_j, \mathcal{M}_i) \leq \varepsilon\}$. Der RANSAC-Algorithmus wird verwendet, um das bestunterstützte bzw. ein hinreichend unterstütztes Muster \mathcal{M} zu finden. Das Muster \mathcal{M}_i ist hinreichend unterstützt, wenn seine Unterstützermenge mindestens t Elemente enthält, wobei t eine problemspezifische Mindestanzahl ist. Entsprechend ist das bestunterstützte Muster das Muster der größten Unterstützermenge. Aus einer Unterstützermenge P_i^* kann durch Ausgleichsrechnung das besteingepasste Muster \mathcal{M}_i^* bestimmt werden.

RANSAC (siehe Algorithmus 1 nach [FB81]) untersucht die Eingabemenge D nicht vollständig auf alle möglichen Muster der Struktur. Auch findet kein systematisches Absuchen eines Rasters möglicher Muster wie bei der Hough-Transformation statt. Statt dessen werden die Parameter der untersuchten Muster \mathcal{M}_i durch zufällig bestimmte $P_i \subset D, |P_i| = k_S$ bestimmt. Dabei ist k_S durch die Struktur \mathcal{S} des gesuchten Musters festgelegt.

Nach einem Durchlauf werden alle Unterstützer des Musters \mathcal{M}^* aus den Eingabedaten entfernt. Das Muster \mathcal{M}^* ist hierbei das Muster, welches durch Ausgleichsrechnung aus der größten Unterstützermenge hervorgegangen ist. Durch wiederholtes Ausführen des Algorithmus 1 können die in D enthaltenen Muster bestimmt werden.

In einem RGB-Streudiagramm der Farbproben in einem typischen Bild der Umgebung zeigen sich den zu findenden Farbklassen entsprechende Ballungen, siehe Abbildung 2.7(a). Diese Ballungen können als Linien oder lang gestreckte Ellipsen betrachtet werden. Diese Beobachtung lässt RANSAC als interessante Methode zu Erhebung zugeordneter Stichproben für die Farbklassifikation erscheinen. In den Eingabedaten, einer Stichprobe nicht zugeordneter Farbproben

2. Subsymbolische Verfahren der Bildverarbeitung

Algorithmus 1: RANSAC

Eingabe : Struktur \mathcal{S} , Abstand $d_R : \mathcal{D} \times \mathcal{S} \mapsto \mathbb{R}$, Eingabemenge D ,
Maximalabstand eines unterstützenden Elements ε , Mindestanzahl der
Unterstützer t , Maximale Iterationszahl n

1 Beginn

/*Unterstützer des bestunterstützten Musters */

2 $P^* \leftarrow \emptyset$

3 $i \leftarrow 0$

4 **solange** $i < n$ **tue**

5 $i \leftarrow i + 1$

6 Bestimme $P_i \subset D, |P_i| = k_{\mathcal{S}}$ durch zufällige Auswahl.

7 Bestimme das Muster \mathcal{M}_i aus P_i .

8 Bestimme die Menge P_i^* aller Unterstützer des Musters \mathcal{M}_i bezüglich
 d_R und ε

9 $P^* \leftarrow P_l^*$ mit $l := \arg \max_{j \in [1, i]} (|P_j^*|)$

10 **wenn** $|P^*| \geq t$ **dann**

11 $i \leftarrow n$

12 \mathcal{M}^* durch Ausgleichsrechnung aus P^* bestimmen.

Ergebnis : \mathcal{M}^*

typischer Bilder der Umgebung, wird nach Mustern der Struktur einer Geraden bzw. Strecke gesucht. Die Anzahl der zu bestimmenden Muster entspricht der Menge der zu unterscheidenden Farbklassen. Die Unterstützer werden ihren jeweiligen Mustern bzw. Farbklassen zugeordnet und stehen somit als Eingabedaten für die Generalisierung zugeordneter Stichproben bereit.

Es wird pro Durchlauf nach einer Strecke gesucht, die aktuelle Strecke nicht unterstützende Punkte werden als Ausreißer gewertet und haben somit keinen Einfluss auf diese. Die Struktur einer Strecke der Endpunkte A, B kann beschrieben werden zu $g : \vec{r}(\lambda) = \vec{a} + \lambda (\vec{b} - \vec{a})$, wobei $\lambda \in [0, 1]$. Mit Hilfe der Hauptkomponentenanalyse (PCA, principal component analysis, [Pea01]) werden die Parameter des Musters \mathcal{M}^* zur Unterstützermenge P^* bestimmt. Hierzu werden die Parameter Schwerpunkt $\vec{\mu}(P^*)$ nach (2.21) und Kovarianzmatrix $\Sigma(P^*)$ nach (2.22) bestimmt, da nach [Pea01] zum einen der arithmetische Mittelwert der Unterstützer Element der Ausgleichsgeraden ist und sich zum anderen der Richtungsvektor der Ausgleichsgeraden als der Vektor ergibt, der die Varianz der Unterstützer maximiert. Dieser varianzmaximierende Vektor ergibt sich als Eigenvektor \vec{E}_{\max} des größten Eigenwertes der Kovarianzmatrix. Zur Bestimmung der Eigenwerte und -vektoren wird nach [PTVF92] eine Householdertransformation [MRW68] mit anschließender QL-Zerlegung durchgeführt. Beide Algorithmen wurden von [PTVF92] übernommen.

Start- und Endpunkt der die Farbklass repräsentierenden Strecke können durch elementweise orthogonale Projektion der Unterstützermenge auf die vorläufige Gerade $h : \vec{r}(\lambda) = \vec{\mu}(P^*) + \lambda \vec{E}_{\max}(\Sigma(P^*))$ bestimmt werden. Die Projektionspunkte auf h mit minimalen bzw. maximalen λ sind Start- bzw. Endpunkt der Strecke.

Mit dem beschriebenen Vorgehen gelingt es nicht, den Farbklassen entsprechenden Partitionierungen zu finden. Eine Ursache für das Scheitern des Verfahrens wird in der Annahme linienartiger Ballungen gesehen. Diese Annahme wird für einige Farbklassen, wie beispielsweise Weiß oder Gelb, gut erfüllt. Bei anderen Farbklassen, z.B. Blau oder bei der Farbe des Hintergrundes, scheint die Linienartigkeit der Verteilungen weniger ausgeprägt. Es wird vermutet, dass die Leistungsfähigkeit des Verfahrens durch die Wahl einer besser geeigneten Struktur der zu bestimmenden Muster (beispielsweise Ellipsoide) sowie durch eine Transformation der Eingabedaten in einen besser geeigneten Farbraum stark verbessert werden kann.

k-Means

K-Means [Mac67] ist ein Verfahren zur Zerlegung einer Menge in paarweise disjunkte Partitionen. Die zu partitionierende Menge \mathfrak{D} unterliegt dabei der Verteilung $P(\vec{x} \in \mathfrak{D}) \mapsto [0, 1] \subset \mathbb{R}$, die zu findende Partitionierung $\mathcal{C} = \{C_1, \dots, C_k\}$ besteht aus den paarweise disjunkten Partitionen bzw. Cluster $C_i \subset \mathfrak{D}$. Nach [Mac67, Lin94, LBG03] wird für die Zerlegung von \mathfrak{D} ein kleiner

Partitionierungsfehler $W^2 = \sum_{i=1}^k \left[\int_{C_i} d_K(\vec{z}, \vec{\mu}_i)^2 dP(\vec{z}) \right]$ angestrebt. Dabei ist $\vec{\mu}_i = E(P(\vec{x} \in C_i))$ der Erwartungswert der Partition C_i und $d_K : \mathfrak{D}^2 \mapsto \mathbb{R}$ die der Zerlegung zugrunde liegende Metrik.

Steht als Eingabedaten eine Stichprobe $D = (\vec{p}_1, \dots, \vec{p}_m)$, $\vec{p}_j \in \mathfrak{D}$ bereit, dann wird ein iteratives Verfahren zur Zerlegung in eine bekannte oder vermutete Anzahl k Cluster realisiert. Die Cluster C_i werden dabei durch Tupel $S_i := (\vec{p}_{i1}, \dots, \vec{p}_{im_i})$ ihnen zugeordneter Glieder aus D repräsentiert. Am Anfang des Verfahrens werden die Erwartungswerte $\vec{\mu}_i$ der Cluster C_i werden zufällig initialisiert. Anfang jedes Durchlaufs werden alle S_i zu $S_i \leftarrow ()$ initialisiert. In einem Durchlauf wird jedes p_j der Eingabedaten dem Cluster C_l des Minimalabstandes hinzugefügt:

$$S_l \leftarrow (\vec{p}_{l1}, \dots, \vec{p}_{lm_l}, \vec{p}_j), \quad l := \arg \min_{i=1}^k d_K(\vec{p}_j, \mu_i). \quad (2.28)$$

2. Subsymbolische Verfahren der Bildverarbeitung

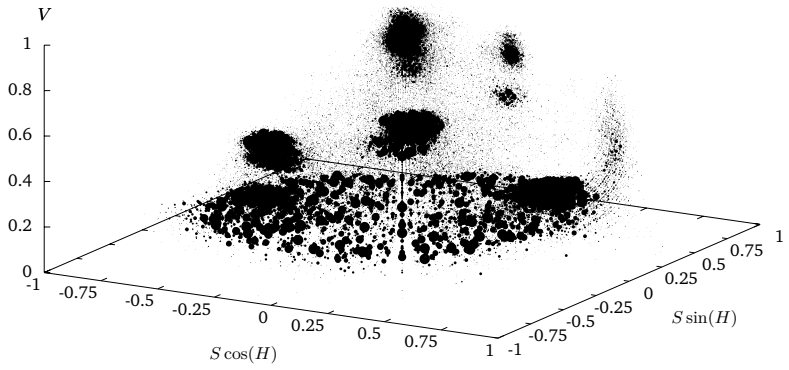
Der Durchlauf wird mit einer Schätzung der Erwartungswerte $\vec{\mu}_i$ der den Clustern C_i zugeordneten Tupeln S_i analog zu (2.21) abgeschlossen.

Als Abbruchkriterien wird eine festgelegte Iterationszahl oder ein sich nicht mehr signifikant ändernder Partitionierungsfehler genutzt. Das Verfahren stellt nicht sicher, dass ein globales Minimum des Partitionierungsfehlers gefunden wird. Bei wiederholter Anwendung des Verfahrens mit jeweils neu zufällig initialisierten Clusterzentren $\vec{\mu}_i$ kann die Partitionierung mit kleinstem verbleibenden Fehler genutzt werden.

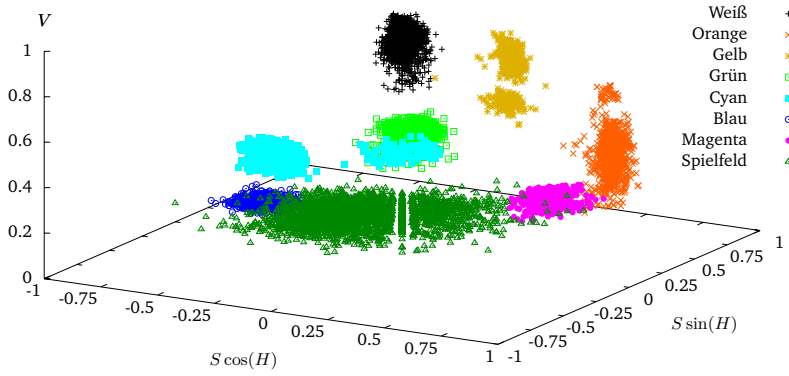
Für den praktischen Einsatz werden $k = |\mathcal{C}|$ und d als euklidische bzw. Mahalanobisdistanz gesetzt. Die Vertreter S_i der Cluster C_i nach dem letzten Durchlauf werden als Partitionierung der Eingabedaten verwendet. Die Eingabedaten werden im RGB-, im HSV- und im HSL-Farbraum [FDSH97] untersucht. Die Untersuchungen im HSV-Farbraum werden sowohl mit und ohne Skalierung der Sättigung auf die Helligkeit durchgeführt, siehe Abbildung 2.11.

Der Verzicht auf diese Skalierung wird durch große Schwankungen der Sättigung bei kleinen Helligkeiten motiviert. Da die stark voneinander abweichenden Auftrittshäufigkeiten der einzelnen Farbklassen zu Problemen bei der Bildung der Partitionen führen könnten, werden zusätzlich verschiedene Quantisierungen der Eingabedaten mit anschließender Unikalisation untersucht. Es können jedoch keine den Farbklassen entsprechenden Partitionierungen erstellt werden. Dies ist ein Indiz dafür, dass die verwendeten Metriken nicht geeignet sind, die Farben der vorhandenen Verteilung zu separieren. Abhilfe hierzu könnten Kernel-Ansätze [CV05] sowie auf der Hauptkomponentenanalyse aufbauende Ansätze [DH04a] bieten. Ein mit k -Means verwandtes Verfahren ist der Fuzzy- C -Means-Algorithmus [BEF84], bei dem die Zuordnung eines Punktes zu einem Cluster nicht diskret sondern kontinuierlich erfolgt. Dieser Aspekt der unscharfen Klassifizierung macht diese Variante ebenfalls interessant für weitere Untersuchungen, eine vorläufige Implementierung für sphärische Cluster versagt jedoch.

2.2. Farbklassifikation

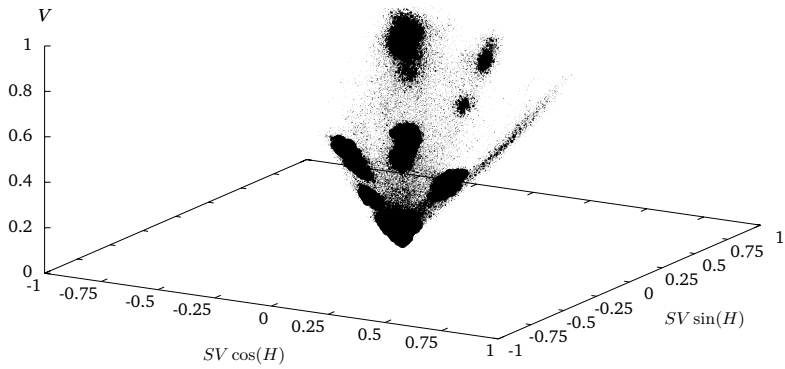


(a)

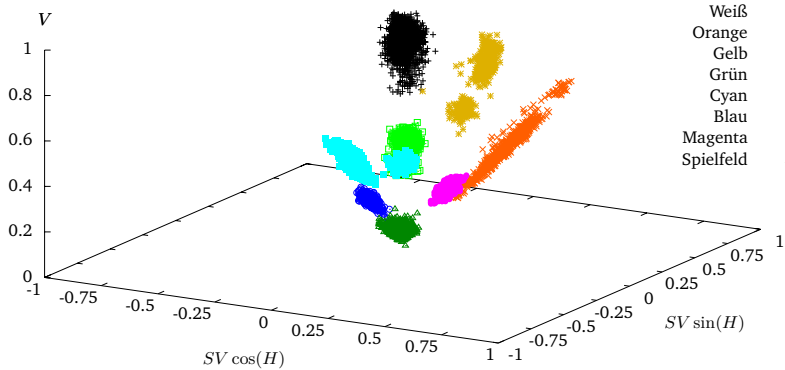


(b)

2. Subsymbolische Verfahren der Bildverarbeitung



(c)



(d)

Abbildung 2.11.: Streudiagramme der auftretenden Farbwerte im HSV-Farbraum (a,b) sowie im HSV-Farbraum ohne Bezug der Sättigung auf die Helligkeit (c,d): unklassifiziert (a,c) und nutzerklassifiziert (b,d).

DBSCAN

Wie k -Means ist auch Density Based Spatial Clustering of Applications with Noise ein Verfahren zur Partitionierung von Eingabedaten D in paarweise disjunkte Teilmengen C_i . Im Unterschied zu k -Means ist hierbei kein Vorwissen über die Anzahl und Gestalt der zu findenden Cluster nötig. Ausgehend von einem Abstand $\mathfrak{d} : \mathfrak{D}^2 \mapsto \mathbb{R}$, einem Maximalabstand ε und einer Mindestmächtigkeit m wird ein Cluster $C \subset D$ der Eingabemenge $D \subset \mathfrak{D}$ im Sinne von DBSCAN in [M. 96] vereinbart durch die Definitionen 1-6. Aufbauend auf der Definition einer ε -Umgebung

Definition 1 Für die ε -Umgebung $N_\varepsilon(p)$ eines Punktes $p \in D$ gilt:

$$N_\varepsilon(p) = \{q \in D \mid \mathfrak{d}(q, p) \leq \varepsilon\}$$

wird dort der Begriff der direkten dichtebasierten Erreichbarkeit eingeführt

Definition 2 Ein Punkt $p \in D$ ist von einem Punkt $q \in D$ aus bezüglich ε und Mindestmächtigkeit der Nachbarschaft m direkt dichtebasiert erreichbar, wenn

1. $p \in N_\varepsilon(q)$ und
2. $|N_\varepsilon(q)| \geq m$

und damit der Begriff der dichtebasierten Erreichbarkeit definiert.

Definition 3 Dichtebasierte Erreichbarkeit zweier Punkte $p, q \in D$ bezüglich ε und m besteht, wenn mindestens ein Pfad p_1, \dots, p_n mit $p_1 = q$ und $p_n = p$ existiert, in dem p_{i+1} von p_i aus bezüglich ε und m direkt dichtebasiert erreichbar ist.

Aus der nicht symmetrischen Eigenschaft der dichtebasierten Erreichbarkeit wird die Eigenschaft der dichtebasierten Verbundenheit abgeleitet

Definition 4 Zwei Punkte $p, q \in D$ sind bezüglich ε und m dichtebasiert verbunden, wenn es einen Punkt $o \in D$ gibt, sodass dichtebasierte Erreichbarkeit bezüglich ε und m sowohl von o zu p als auch von o zu q besteht.

und hiermit die Definition eines Clusters formuliert:

Definition 5 Ist D eine Menge von Punkten, dann ist $C \subset D, C \neq \emptyset$ ein Cluster bezüglich ε und m , wenn die folgenden Bedingungen erfüllt sind:

1. $\forall p, q$: Wenn $p \in C$ und q bezüglich ε und m von p aus dichtebasiert erreichbar ist, dann $q \in C$.

2. Subsymbolische Verfahren der Bildverarbeitung

2. $\forall p, q \in C$: p ist dichtebasiert verbunden mit q bezüglich ε und m .

Weiterhin wird der Begriff des Rauschens bereitgestellt:

Definition 6 Sind C_1, \dots, C_k Cluster der Punktmenge D bezüglich ε_i und m_i , dann ist
 $Rauschen = \{p \in D \mid \forall i \in [1, k] : p \notin C_i\}$

In [M. 96] wird der Algorithmus DBSCAN (Algorithmus 3) zum Finden aller Cluster C_i der Menge D entsprechend Definition 5 vorgestellt. Hierzu wird die Expansion eines Clusters nach Algorithmus 2 [M. 96] verwendet. Das Interessante an der zugrunde liegenden Definition ist, dass ein Cluster lediglich durch eine Abstandsdefinition und die Beschreibung lokaler ε -Nachbarschaft charakterisiert wird.

Algorithmus 2: ExpandiereCluster

Eingabe : Daten D , Punkt p , Nachbarschaft N , Cluster C ,
 Umgebungsbeschreibung (ε, m)

```

1 Beginn
2    $C \leftarrow C \cup \{p\}$ 
3   für jedes  $p_i \in N$  tue
4     wenn  $p_i$  unbesucht dann
5       Markiere  $p_i$  als besucht
6       Nachbarschaft  $N_i \leftarrow \text{Bereichssuche}(D, p_i, \varepsilon)$ 
7       wenn  $|N_i| \geq m$  dann
8          $N \leftarrow N \cup N_i$ 
9     wenn  $p_i$  zu keinem Cluster gehört dann
10       $C \leftarrow C \cup \{p_i\}$ 

```

Für die Anwendung des DBSCAN-Algorithmus wird die zu partitionierende Stichprobe in einen den RGB-Farbraum repräsentierenden Akkumulator übertragen, welcher eine einfache Realisierung von Bereichsanfragen gestattet. Werden in diesem Akkumulator die Realisierungen der Elemente des Farbraumes durch die Stichprobe markiert, so bildet die Menge aller markierten Einträge die Eingabedaten des Verfahrens. Die relativen Häufigkeiten der einzelnen Farbwerte werden somit nicht bei der Auswertung berücksichtigt. Hiermit gelingt es nicht,

Algorithmus 3: DBSCAN

```

Eingabe : Daten  $D$ , Umgebungsbeschreibung  $(\varepsilon, m)$ 
1 Beginn
2    $C \leftarrow \emptyset$ 
3   für alle unbesuchten Punkte  $p \in D$  tue
4     Markiere  $p$  als besucht
5      $N \leftarrow \text{Bereichssuche}(D, p, \varepsilon)$ 
6     wenn  $|N| < m$  dann
7       Markiere  $p$  als Rauschen
8     sonst
9        $C \leftarrow$  nächster Cluster
10       $\text{ExpandiereCluster}(D, p, N, C, \varepsilon, m)$ 

```

in den vorliegenden Eingabedaten Partitionen zu finden. Werden mit dem Akkumulator jedoch die relativen Häufigkeiten der Farbwerte in der Stichprobe abgebildet⁵, dann lassen sich mit dem Verfahren Partitionen erstellen. Dabei wurde bisher keine Beschreibung der lokalen ε -Nachbarschaft gefunden, die eine den vorkommenden Farbklassen entsprechende Partitionierung erzeugt. Als Ursache hierfür werden die großen Unterschiede der relativen Häufigkeiten der einzelnen Farbklassen in der Stichprobe gesehen. Insbesondere die Farbe Orange ist in den Eingabedaten unzureichend repräsentiert, siehe Abbildung 2.7(a). Durch Hinzufügen einer nutzerklassifizierten Stichprobe für Orange konnte jedoch eine den vorkommenden Farbklassen entsprechende Partitionierung erzeugt werden, siehe Abbildung 2.12. Dieses Ergebnis legt den Schluss nahe, dass die Bereitstellung der Eingabedaten in Zukunft entsprechend Abbildung 2.13 vereinfacht werden kann, wobei ausreichend ausgeglichene relative Häufigkeiten der einzelnen Farbklassen sicherzustellen sind.

⁵Dies ist ohne Widerspruch mit den Definitionen 1-6 durch Indizierung mit $V \subset \mathbb{N}$,
 $x := (r_x, g_x, b_x, i_x) \in \mathfrak{D} = V^3 \times \mathbb{N}$ und $\mathfrak{D}^2 \ni (x, y) \mapsto \mathfrak{d}(x, y)$
 $= \sqrt{(r_x - r_y)^2 + (g_x - g_y)^2 + (b_x - b_y)^2}$ zu erreichen, wenn dafür gesorgt wird, dass jedes
 Glied der Stichprobe anhand des Index i unterschieden werden kann.

2. Subsymbolische Verfahren der Bildverarbeitung

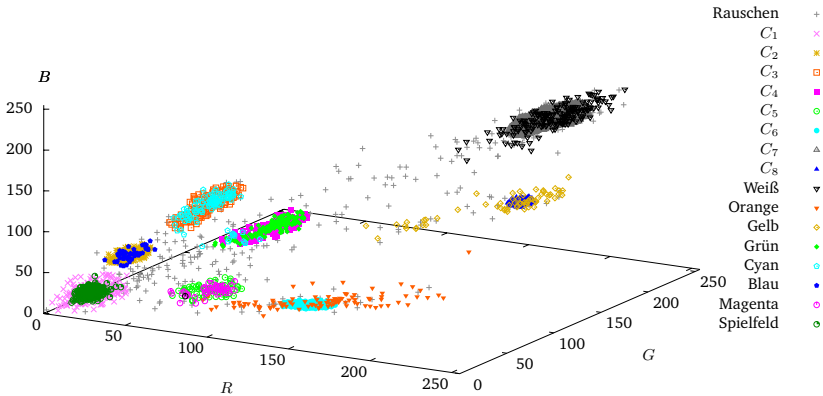


Abbildung 2.12.: Gegenüberstellung von nutzerklassifizierten Stichproben nach Abbildung 2.7(b) und durch den DBSCAN-Algorithmus bestimmten Partitionen (jeweils in komplementärer Farbe dargestellt).

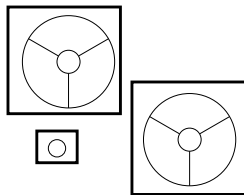


Abbildung 2.13.: Vereinfachte Bereitstellung der Eingabedaten durch grobes Markieren

2.3. Segmentierung

Ziel der Segmentierung ist es, im Bild Bereiche gleicher Merkmalsausprägung zu identifizieren. Bei der Anwendung der Bildverarbeitung zur Objektlokalisierung im RoboCup hat die Segmentierung die Aufgabe, die Markierungen der Roboter sowie den Ball im Videobild zu detektieren. Aus dem Vorwissen über das System (geometrisches Modell der Robotermarkierungen, Abbildungsmaßstab des optischen Systems) kann mit Hilfe der Ergebnisse der Segmentierung Wissen über den Zustand der Spielumgebung gewonnen werden. Als Segment wird eine zusammenhängende Menge von Bildelementen gleicher Farbkategorie verstanden. Ein Segment soll dabei maximal expandiert sein, also keine direkten Nachbarn gleicher Farbkategorie außerhalb dieser Menge haben. Diese Eigenschaften eines Segmentes legen nahe, ein Segment als DBSCAN-Cluster bezüglich Definition 5 zu verstehen.

Die Eingangsdaten des Segmentierungsverfahrens $\mathbf{D} \in \mathbb{C}^{M \times N}$ liegen als örtlich referenzierte Matrix eindeutiger Klassenzugehörigkeiten vor. Wird diese Matrix als Menge D von Bildelementen $p_i = (y_i, x_i, c_i) = (y_i, x_i, \mathbf{D}_{y_i, x_i}) \in D \subset \mathfrak{D}$ interpretiert, dann führt der Abstand d_P zwischen Bildelementen

$$d_P : \mathfrak{D}^2 \rightarrow \mathbb{R}$$

$$(p_a, p_b) \mapsto d_P(p_a, p_b) = \begin{cases} \varepsilon + 1, & \text{wenn } c_a \neq c_b \\ |x_b - x_a| + |y_b - y_a|, & \text{sonst} \end{cases} \quad (2.29)$$

zu Definition der Segmente im Sinne vorliegender Arbeit:

Definition 7 Sei D die Menge aller Elemente eines Bildes, so sind bezüglich des Abstandes (2.29), des Maximalabstandes $\varepsilon := 1$ und der Mindestmächtigkeit der ε -Umgebung $m = |N_\varepsilon(p)| := 1$ alle DBSCAN-Cluster C_i nach Definition 5 Segmente.

2. Subsymbolische Verfahren der Bildverarbeitung

Durch Festlegung der Mindestmächtigkeit $m = |N_\varepsilon(p)| := 1$ einer ε -Umgebung verschwindet die Eigenschaft der Dichtebasiertheit in den Definitionen 2-5. Zwei Bildelemente p, q gleicher Klassifikation sind dann direkt erreichbar nach Definition, wenn sie Von-Neumann-Nachbarn sind. Im bisherigen System[Ler08] wurden beim Aufbau der Segmente für jedes Bildelement zwei Verbundenheitstests durchgeführt (Abbildung 2.3). Es wird deutlich, dass diese Tests zum Teil redundant sind.

Ein alternatives Verfahren besteht in der zeilenweisen Lauflängenkodierung der Klassenzugehörigkeiten und anschließender Vereinigung benachbarter Gruppen gleicher Klassenzugehörigkeiten zu Segmenten nach [BBV00, MDSG02].

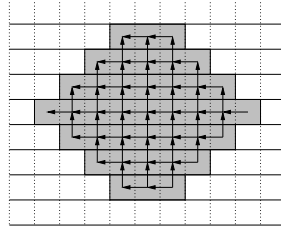


Abbildung 2.14.: Bisheriges Segmentierungsverfahren

2.3.1. Lauflängenkodierung

Die Lauflängenkodierung ist ein Verfahren der Datenkompression. Ihr Einsatz zur Segmentierung wird durch die Einsparung redundanter Verbundenheitstests motiviert.

Das klassifizierte Bild \mathbf{D} wird zeilenweise in Gruppen

$$G_i := (y_i, x_i, w_i, c_i, s_i, r_i) \quad (2.30)$$

zerlegt. Dabei bilden (y, x, w, c) eine komprimierte Beschreibung der dichtebasiert verbundenen Bildelemente $\{(y, x, c), (y, x + 1, c), \dots, (y, x + w - 1, c)\}$. Der Bezeichner s und der Bezeichner der Wurzelgruppe r werden zur Bildung der Segmente aus Gruppen benötigt.

Die Menge aller Gruppen der Zeile i sei L_i , die Menge aller Gruppen des Bildes $\mathcal{G} = \{L_1, \dots, L_M\}$. Die Menge \mathcal{G} wird zeilenweise durch Algorithmus 4 aufgebaut.

Hierbei stehen noch keine Nachbarschaftsinformationen über Zeilen hinweg zur Verfügung. Durch die Initialisierungen $s(G_i) \leftarrow i$ und $p(G_i) \leftarrow s(G_i)$ bildet jede Gruppe die Wurzel eines neuen Segmentes. Das Ergebnis dieses Vorgehens nach [BBV00] ist ein vollständig disjunkter Wald (vergleiche [Die06]). Beim anschließenden Verfahren der Segmentvereinigung werden unter Nutzung der Nachbarschaftsinformationen über Zeilen hinweg die Bäume des Waldes bestimmt. Diese Bäume entsprechen den zu bestimmenden Segmenten, ihre Elemente kodieren alle am jeweilige Segment beteiligten Bildelemente.

Algorithmus 4: Lauflängenkodierung

```

Eingabe : Klassifiziertes Bild  $\mathbf{D} \in \mathbb{C}^{M \times N}$ 
Ausgabe :  $\mathcal{G} := \{L_1, \dots, L_M\}$ 
1 Beginn
2    $\mathcal{G} \leftarrow \emptyset, s \leftarrow 1$ 
3   für jedes  $i \in [1, M]$  tue
4     /*Gruppen der Zeile  $i$  */
5      $L_i \leftarrow \emptyset$ 
6     /*Gruppe der Spalte  $j_c$ , der Breite  $w$ , der Klasse  $c_c$  und des
       Bezeichners  $s$  öffnen */
7      $j_c \leftarrow 1, w \leftarrow 0, c_c \leftarrow \mathbf{D}_{i,1}$ 
8     für jedes  $j \in [2, N]$  tue
9        $w \leftarrow w + 1$ 
10      wenn  $c_c \neq \mathbf{D}_{i,j}$  dann
11        /*Gruppe finalisieren */
12         $L_i \leftarrow L_i \cup \{(i, j_c, w, c_c, s, s)\}$ 
13        /*neue Gruppe der Klasse  $C_c$  und der Spalte  $j_c$  öffnen */
14         $j_c \leftarrow j, w \leftarrow 0, c_c \leftarrow \mathbf{D}_{i,j}, s \leftarrow s + 1$ 
15      /*letzte Gruppe finalisieren */
16       $L_i \leftarrow L_i \cup \{(i, j_c, w, c_c, s, s)\}$ 
17       $s \leftarrow s + 1$ 
18       $\mathcal{G} \leftarrow \mathcal{G} \cup \{L_i\}$ 

```

2.3.2. Segmentvereinigung

Direkte Erreichbarkeit zweier Gruppen G_a, G_b bzw. Verbundenheit aller durch sie kodierten Elemente entsprechend Definition 7 besteht, wenn

G_b ist direkt erreichbar von $G_a \iff$

$$(c_a = c_b) \wedge (|y_b - y_a| = 1) \wedge (x_a \leq x_b + w_b - 1) \wedge (x_b \leq x_a + w_a - 1) \quad (2.31)$$

Zur Vereinigung zweier direkt erreichbarer Gruppen G_a, G_b gleicher Klassifikation steht Algorithmus 5 zur Verfügung. Hierbei werden bei spät erkannter Äquivalenz zweier Segmente jeweils nur die Bezeichner der jeweiligen Wurzelgruppen aktualisiert. Algorithmus 6 bildet die einzelnen in D enthaltenen Segmente durch zeilenweise Vereinigung erreichbarer Gruppen gleicher Farbklasse, Abbildung 2.15 zeigt hierzu ein Beispiel. Nach der Vereinigung der erreichbaren Gruppen gilt für alle Segmente $C_i \subset D$

$$C_i = \{G_j \in L_k \in \mathcal{G} \mid s(G_{r(G_j)}) = i\}. \quad (2.32)$$

2. Subsymbolische Verfahren der Bildverarbeitung

Algorithmus 5: Vereinige

```

Eingabe :  $G_a, G_b$ 
1 Beginn
   /*Orientierung am Bezeichner der Wurzelgruppe */
2   wenn  $s(G_r(G_a)) < s(G_r(G_b))$  dann
   /*Bezeichner der Wurzelgruppe aktualisieren */
3    $s(G_r(G_b)) \leftarrow s(G_r(G_a))$ 
   /*Wurzelgruppe aktualisieren */
4    $r(G_b) \leftarrow r(G_a)$ 
5   sonst
6   | Vereinige( $G_b, G_a$ )

```

Die Anzahl A_{C_i} der Bildelemente des Segments C_i zu

$$A_{C_i} = \sum_{G \in C_i} [w(G)] \quad (2.33)$$

entspricht über die Geometrie der Bildelemente einem Flächeninhalt. Mit ihr kann der Segmentsschwerpunkt bezüglich des Sensorkoordinatensystems zu

$$\vec{r}_{C_i} = \frac{1}{2A_{C_i}} \sum_{G \in C_i} \left[w(G) \begin{pmatrix} x(G) + w(G) - 1 \\ 2y(G) \end{pmatrix} \right] \quad (2.34)$$

bestimmt werden. Aus den Segmentsschwerpunkten kann auf die Lage der Segmente in zur Bildebene parallelen Objektebenen geschlossen werden.⁶ In Tabelle 2.1 sind die mit dem beschriebenen Verfahren erzielten Leistungskennwerte dargestellt.

⁶Die hierfür nötige Kalibrierung erfolgt momentan rudimentär: Die Kamera wird mit Hilfe visueller Kontrolle der Spielfeldkanten im Videobild parallel zum Spielfeld ausgerichtet und der Abbildungsmaßstab in der Spielfeldebene wird durch die Ausdehnung des Spielfeldes im Videobild ermittelt. Die Entfernung der Kamera von der Spielfeldebene wird durch Messung bestimmt. Es wird jedoch eine Verbesserung der Genauigkeit durch Einsatz von Methoden der Kamerakalibrierung nach [HZ03] erwartet.

Algorithmus 6: Segmentierung

```

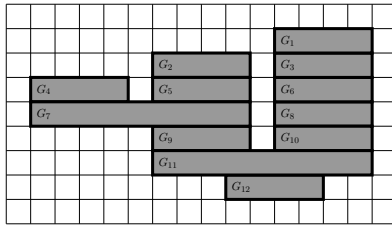
Daten :  $\mathcal{G} = \{L_1, \dots, L_M\}$ 
1 Beginn
2   für jedes  $l \in [2, M]$  tue
3     /*Gruppen der letzte Zeile  $A$ , aktives Element  $i$  */
4      $A \leftarrow L_{l-1}$ 
5      $i \leftarrow 1$ 
6     /*Gruppen der aktuellen Zeile  $B$ , aktives Element  $j$  */
7      $B \leftarrow L_l$ 
8      $j \leftarrow 1$ 
9     solange  $(i \leq |A|) \wedge (j \leq |B|)$  tue
10      wenn  $x(B_j) > x(A_i) + w(A_i) - 1$  dann
11        /* $A_i$  endet vor dem Start von  $B_j$ , keine
12          Erreichbarkeit, nächstes Element in  $A$  aktivieren
13          */
14         $i \leftarrow i + 1$ 
15      sonst wenn  $x(A_i) > x(B_j) + w(B_j) - 1$  dann
16        /* $B_j$  endet vor dem Start von  $A_i$ , keine
17          Erreichbarkeit, nächstes Element in  $B$  aktivieren
18          */
19         $j \leftarrow j + 1$ 
20      sonst
21        wenn  $c(A_i) = c(B_j)$  dann
22          /* $A_i$  und  $B_j$  sind erreichbar nach (2.31) */
23          Vereinige( $A_i, B_j$ )
24          /*Aktive Elemente aktualisieren */
25          wenn  $x(B_j) + w(B_j) > x(A_i) + w(A_i)$  dann
26             $i \leftarrow i + 1$ 
27          sonst
28             $j \leftarrow j + 1$ 

```

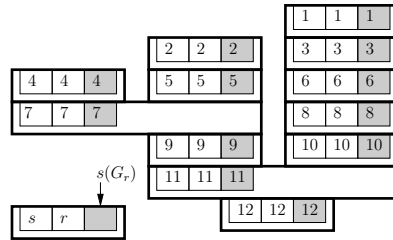
Vorgabe	Bearbeitungszeit ≤ 40 ms für vollständige Auswertung eines Bildes
Bearbeitungszeit	25 ms pro Bild für vollständige Auswertung mit Bayerinterpolation, Helligkeitsausgleich, Farbklassifikation und Segmentierung
Bildausschnitt	$1,5 \cdot 10^6$ Bildelemente
Klassifikatorauflösung	5 Bit pro Farbkanal
Prozessor	Intel®Core™2 Duo CPU T8100, 2,1GHz

Tabelle 2.1.: Leistungskennwerte des Segmentierungsverfahrens

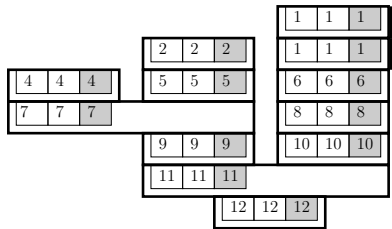
2. Subsymbolische Verfahren der Bildverarbeitung



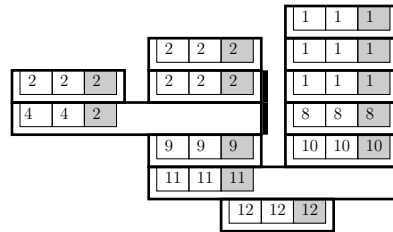
(a) Bei der Lauflängenkodierung wurden die Gruppen G_1, \dots, G_{12} gleicher Farbkasse in einem Bild gefunden. Sie werden durch Algorithmus 6 zu einem Segment vereinigt. Am Beispiel wird die Vereinigung erreichbarer Gruppen nach Algorithmus 5 gezeigt.



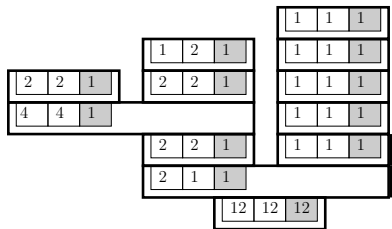
(b) Initialisierung der Segmentbezeichner (jeweils erstes dargestelltes Feld in einer Gruppe) zu $s(G_i) \leftarrow i$ und der Bezeichner der Wurzelgruppen (jeweils zweites dargestelltes Feld in einer Gruppe) zu $p(G_i) \leftarrow s(G_i)$. Im grau hinterlegten Feld ist jeweils der Bezeichner $s(G_{r(G_i)})$ der Wurzelgruppe dargestellt.



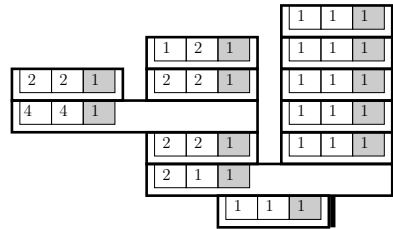
(c) Die schwarze Markierung zeigt die Vereinigung der Gruppen G_1 und G_3 an. Da $s(G_{r(G_1)}) = s(G_1) = 1 < s(G_{r(G_3)}) = s(G_3) = 3$, wird G_1 zur neuen Wurzelgruppe von G_3 . $s(G_{r(G_3)}) \leftarrow s(G_{r(G_1)})$, $r(G_3) \leftarrow r(G_1)$. Das Bild zeigt den Zustand nach der Vereinigung.



(d) Vereinigung der Gruppen G_5 und G_7 : Da $s(G_{r(G_5)}) = s(G_2) = 2 < s(G_{r(G_7)}) = s(G_4) = 4$, wird G_2 zur neuen Wurzelgruppe von G_4 . $s(G_{r(G_5)}) \leftarrow s(G_{r(G_7)})$, $r(G_5) \leftarrow r(G_7)$. Die vorherigen Werte $s(G_7)$ und $p(G_7)$ werden nicht verändert. Bereits durch Veränderung der Wurzelgruppe ist $s(G_{r(G_7)})$ valide.



(e) Vereinigung der Gruppen G_{10} und G_{11} : Da $s(G_{r(G_{10})}) = s(G_1) = 1 < s(G_{r(G_{11})}) = s(G_2) = 2$, wird G_1 zur neuen Wurzelgruppe von G_2 . $s(G_{r(G_{11})}) \leftarrow s(G_{r(G_{10})})$, $r(G_{11}) \leftarrow r(G_{10})$. Die Zugehörigkeit des linken Zweiges klappt auf Segment 1 um.



(f) Das Segment 1 wurde vollständig erkannt. Jedes Segment $S_i \in D$ ist gegeben durch $C_i = \{G_j \in \mathcal{G} \mid s(G_{r(G_j)}) = i\}$.

Abbildung 2.15.: Beispiel der Vereinigung erreichbarer Gruppen nach Algorithmus 6.

3. Symbolische Auswertung – Bildinterpretation

3.1. Trikots

3.1.1. Anforderungen

Die Lokalisation der Spieler geschieht in der RoboCup Small Size League durch über dem Spielfeld angeordnete Kameras. Die Spieler sind farblich markiert, um sowohl Position und Ausrichtung als auch Mannschaftszugehörigkeit und Spielernummer bestimmen zu können. Die Spielfeldgröße beträgt ca. $6\text{m} \times 4\text{m}$, der Maximalradius eines Spielers beträgt 90mm und seine für die Kamera sichtbare Fläche somit ca. $0,025\text{m}^2$. Alle Spieler zusammen bedecken also nur ca. 1% des Spielfeldes.

Die Anordnung der Markierungen auf einem Spieler bildet ein Muster. Aus diesem Muster, dem Trikot des Spielers, müssen die Spielernummer und die Ausrichtung des Roboters eindeutig bestimmbar sein. Die Markierungen werden durch die Segmentierung als Segmente erkannt. Nach der Zuordnung dieser Segmente zu den einzelnen Markierungen können Spielernummer und Orientierung bestimmt werden. Zur Kodierung der Mannschaftszugehörigkeit eines Spielers sind die Farben Gelb und Blau vorgesehen, mit den Farben Grün, Cyan, Magenta und Weiß dürfen Spielernummer und Ausrichtung kodiert werden, siehe Abschnitt 2.2.

3.1.2. Kodierung

Bei allen Mustern ist ein Zentralkreis reservierter Farbe zur Kodierung der Mannschaftszugehörigkeit vorhanden. Für die Zuordnung der außenliegenden Segmente bietet sich die Auswertung der relativen Lage zum Zentralsegment an. Die Länge des Vektors vom Zentralsegment auf ein Außensegment ermöglicht eine Zuordnung zu Suchkreisen. Die paarweise Auswertung einschließender Winkel dieser Vektoren ermöglicht es, die Segmente relativ zu ordnen [Ler08]. Die Zuordnung der Außensegmente zu den Markierungen im Muster erfolgt durch die Auswertung eines Startmerkmals. Abbildung 3.1 zeigt verschiedene Realisierungen von Startmerkmalen bzw. Kodierungsansätzen. Bei Variante

3. Symbolische Auswertung – Bildinterpretation

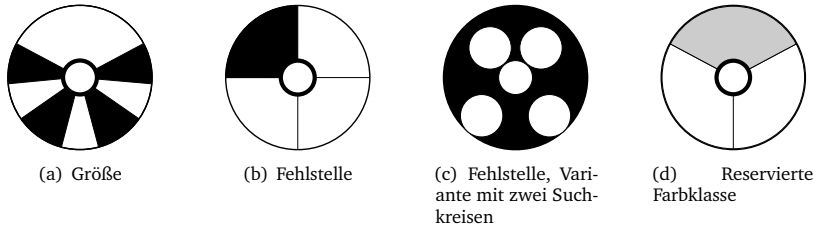


Abbildung 3.1.: Verschiedene Startmerkmale

3.1(a) nach [BBD⁺09] ist eine Markierung signifikant größer als die anderen. Die unterschiedliche Ausprägung der drei kleinen weißen Segmente auf verschiedenen Spielern deutet eine binäre Kodierung des Wortes an.¹ Die Variante 3.1(b) nutzt eine signifikante Lücke (nicht klassifizierte Bildelemente), 3.1(d) ein Segment einer reservierten Farbkasse als Startmerkmal. Diese beiden Varianten sind sehr ähnlich, jedoch wird bei 3.1(b) die verfügbare Fläche zugunsten geringerer Anforderungen an die Leistungsfähigkeit der Farbklassifikation schlechter genutzt. Das aktuelle Standardmuster 3.1(c) der RoboCup Small Size League verfügt über zwei Suchkreise. Über die eingeschlossenen Winkel der Segmente eines Suchkreises bezüglich der Zentralmarkierung sind die einzelnen Markierungen eindeutig identifizierbar. Um möglichst gute Flächenausnutzung zu erreichen, liegen die benachbarten Markierungen des einen Suchkreises in der Fehlstelle des anderen. Mit zunehmender Differenz der Suchkreisradien verschlechtert sich einerseits die Flächenausnutzung, kann aber andererseits ein Segment robuster dem Innen- oder Außenkreis zugeordnet werden. Im Grenzfall, dass die Differenz der Suchkreisradien verschwindet, können bei einem idealisiertem kreisrunden Muster mit Außenradius $r_R = 90\text{mm}$ und Radius der Zentralmarkierung $r_A = 25\text{mm}$ die Radien der Außenmarkierungen r_M noch genau $(r_R - r_A)/2 = 32,5\text{mm}$ betragen. Damit ergibt sich die Grenze des Flächennutzungsgrad zu $A_{\text{nutz}}/A_{\text{gesamt}} = (\pi r_A^2 + 4\pi r_M^2)/(\pi r_R^2) \approx 0,6$. Abbildung 3.2(links) zeigt das bisherige Trikot der Spieler am Fachgebiet nach diesem Schema mit dem schlechten Flächennutzungsgrad $A_{\text{nutz}}/A_{\text{gesamt}} \approx 0,2$.

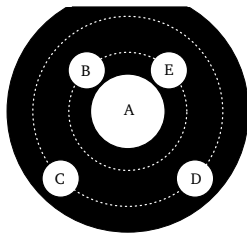
3.1.3. Festlegung eines Trikots

Am RoboCup Versuchsstand der TU Ilmenau wurde bisher ein Trikot nach [Ler08] (siehe Abbildung 3.2) verwendet. Diese Anordnung entspricht der

¹Laut [BBD⁺09] werden die Markierungen nicht segmentbasiert sondern kantenbasiert ausgewertet.

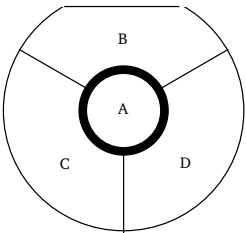
Variante nach Abbildung 3.1(c), die Spielernummer ist dabei binär in den Außenmarkierungen kodiert. Der Hauptkritikpunkt an dieser Anordnung ist die schlechte Ausnutzung der verfügbaren Fläche. Da nur wenige auswertbare Bildelemente zur Verfügung stehen, hat das Messrauschen erheblichen Einfluss auf die Genauigkeit der Lagebestimmung. Dabei erweist sich die Bestimmung der Position als ausreichend robust, die Genauigkeit der Ausrichtungsbestimmung ist jedoch unbefriedigend. Ein verbreiteter Ansatz zur Lösung dieses Problems in der RoboCup Small Size League ist der Einsatz von Beobachtern zur modellgestützten Messwertaufbereitung. Es wird vorgeschlagen, Verfahren dieser Klasse in Zukunft zur Messwertaufbereitung für das RoboCup-Projekt der TU Ilmenau zu verwenden. Vorerst soll das Trikot im Sinne besserer Flächennutzung verändert werden, um die Auswirkungen stochastischer Effekte der Bildaufnahme auf die Genauigkeit der Lagebestimmung zu reduzieren. Deshalb wird das Trikot entsprechend Variante 3.1(d) festgelegt, siehe Abbildung 3.2(rechts). Zwei Nachteile gegenüber dem bisherigen Trikot ergeben sich aus dieser Festlegung: Zum einen kann es Probleme bei der Auflösung eng benachbarter Spieler geben, da das Trikot keinen ungenutzten äußeren Rand besitzt. Zwei gleichfarbige Markierungen können so bei der Bildverarbeitung zu einem Segment verschmelzen. Dieses Problem wird softwareseitig durch eine suchkreisbasierte Auswertung der Segmente gelöst. Der zweite Nachteil ist die erhöhte Anforderung an das Auflösungsvermögen des Farbklassifikators, da zwei zusätzliche Farbklassen zur Kodierung der Spielernummer benötigt werden. Diese Aufgabe wird jedoch unter den aktuellen Bedingungen bewältigt.

3. Symbolische Auswertung – Bildinterpretation



Die Marken *B* und *E* liegen auf einem inneren, die Marken *C* und *D* auf einem äußeren Suchkreis. Somit sind die von der Bildverarbeitung detektierten Segmente den einzelnen Marken robust zuzuordnen. Mit Hilfe einer binären Kodierung lassen sich pro Mannschaft 2^4 Spieler zuordnen. Da hierfür nur zwei Farbklassen aus $\{W, G, C, M\}$ benötigt werden, ergeben sich nur geringe Anforderungen an das Auflösungsvermögen der Farbklassifikation. Durch die großen Fehlflächen wird sichergestellt, dass die Markierungen verschiedener Roboter isolierte Segmente bilden, welchen den jeweiligen Trikots sicher zugeordnet werden können.

Nr.	A	B	C	D	E
0	Y B	W	W	W	W
1	Y B	W	W	W	G
2	Y B	W	W	G	W
⋮	⋮	⋮	⋮	⋮	
14	Y B	G	G	G	W
15	Y B	G	G	G	G



Hauptkritikpunkt der alten Trikotvariante ist die schlechte Ausnutzung der verfügbaren Fläche. Bei der neuen Variante hingegen wird die verfügbare Fläche nahezu vollständig genutzt. Dies entspricht bei ansonsten gleichen Bedingungen einem Zuwachs auswertbarer Bildelemente um Faktor 5. Der Frontmarke *B* ist immer die Farbklassse Weiß zugeordnet, hierdurch können die Marken *C* und *D* sicher identifiziert werden. Es werden 2 zusätzliche Farbklassen benötigt, um mindestens 5 Spieler zu adressieren. Hierdurch steigen die Anforderungen an das Auflösungsvermögen des Farbklassifikators.

Nr.	A	B	C	D
0	Y B	W	C	M
1	Y B	W	M	C
2	Y B	W	C	G
3	Y B	W	G	C
4	Y B	W	M	G
5	Y B	W	G	M

Abbildung 3.2.: Vergleich der Trikotvarianten alt(links) und neu(rechts) sowie der verwendeten Kodierungen mit Gelb(Y), Blau(B), Weiß(W), Grün(G), Cyan(C) und Magenta(M). Die schwarzen Flächen sind bei der Bildverarbeitung nicht berücksichtigte Fehlflächen.

3.2. Lokalisation

3.2.1. Helmert-Transformation

Ein Grundproblem des Vermessungswesens ist die Bestimmung von Objektpositionen für die Kartenerstellung mit Hilfe von Methoden der Entfernungs- und Winkelmessung. Hierzu wird zunächst die Lage bekannter Fixpunkte im zu bestimmenden lokalen Koordinatensystem des Vermessers (Beobachtersystem) ermittelt. Aus den Positionen dieser Fixpunkte im Bezugs- oder Referenzsystem kann auf Lage und Ausrichtung des lokalen Systems geschlossen werden. Hierdurch wird es möglich, die Positionen neu vermessener Objekte aus dem Beobachter- in das Referenzsystem zu übertragen. Wären die Positionen der Fixpunkte sowohl im Referenz- als auch im Beobachtersystem exakt bestimmt, so könnte für ebene Probleme bereits aus zwei Fixpunkten auf das lokale System geschlossen werden. Da reale Messungen fehlerbehaftet sind, wird ein Verfahren der Ausgleichsrechnung zur Bestimmung des Beobachtersystems verwendet. Dieses Verfahren wird als Helmert-Transformation bezeichnet. Der Ansatz des Verfahrens ist die Minimierung der Summe der Restklaffungen, der Beträge der Positionsfehler der Fixpunkte in Referenzsystem und transformiertem Beobachtersystem. Es existieren Varianten des Verfahrens sowohl für ebene als auch für räumliche Probleme.

Abbildung 3.3 zeigt, wie die ebene Variante des Verfahrens im Kontext der digitalen Bildverarbeitung zur Bestimmung von Spielerposition und -ausrichtung genutzt werden kann: Den Ortsvektoren \vec{r}_i im Referenzsystem (\vec{e}_1, \vec{e}_1) entsprechen die Ortsvektoren \vec{s}_i im Beobachtersystem (\vec{E}_1, \vec{E}_2). Dabei sind den \vec{r}_i Messfehler und den \vec{s}_i Fertigungsfehler überlagert. Es wird nun eine Transformation aus Drehstreckung und Translation gesucht, die beide Systeme mit minimalem Fehler ineinander abbildet. Die Transformation hat die Form

$$m \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} (\vec{s}_i - \vec{s}_0) = (\vec{r}_i - \vec{r}_0) + \vec{\varepsilon}_i. \quad (3.1)$$

Die Transformationsparameter sind Maßstab m , Orientierung des Beobachtersystems als zwischen \vec{e}_1 und \vec{E}_1 eingeschlossener Winkel φ , Beobachtertranslation \vec{s}_0 , Referenztranslation \vec{r}_0 und Fehlervektor der Vermessung eines einzelnen Fixpunktes $\vec{\varepsilon}_i$. Den Ortsvektoren auf die Fixpunkte \vec{s}_i im Beobachtersystem entsprechen dabei die jeweils zugehörigen Ortsvektoren \vec{r}_i im Referenzsys-

3. Symbolische Auswertung – Bildinterpretation

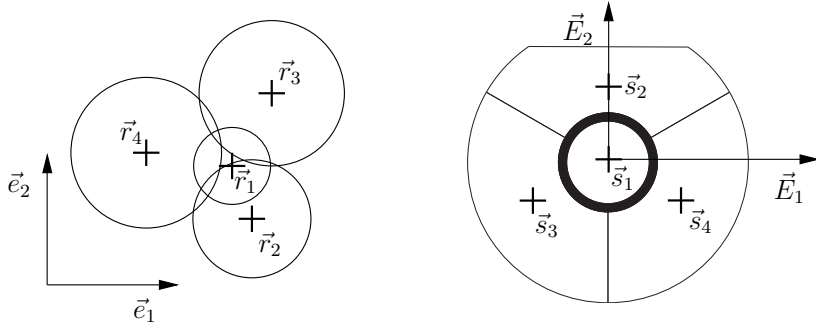


Abbildung 3.3.: Ergebnis der digitalen Bildverarbeitung sind die Segment-schwerpunkte im Referenzkoordinatensystem (\vec{e}_1, \vec{e}_2) sowie die Segmentflächeninhalte (links). Aus einem Trikotmodell können die Markierungsschwerpunkte im Beobachterkoordinatensystem (\vec{E}_1, \vec{E}_2) sowie die Markierungsflächeninhalte bestimmt werden (rechts). Mit Hilfe der Helmert-Transformation werden Ursprung und Orientierung des Beobachtersystems im Referenzsystem bestimmt.

tem. Hiermit lässt sich das zu minimierende Gütekriterium, die Summe der verbleibenden Einzelfehler, aufstellen zu:

$$E = \sum_{i=1}^n [\vec{\varepsilon}_i^2] \rightarrow \min. \quad (3.2)$$

Diese Aufgabe wird nach [GJ07] durch die Gleichungen 3.3 bis 3.6 gelöst, eine Herleitung hierzu findet sich in [Vol03]. Der Translation \vec{s}_0 der Fixpunkte im Beobachtersystem

$$\vec{s}_0 = \frac{1}{n} \sum_{i=1}^n [\vec{s}_i] \quad (3.3)$$

entspricht die Translation \vec{r}_0 der Fixpunkte im Referenzsystem

$$\vec{r}_0 = \frac{1}{n} \sum_{i=1}^n [\vec{r}_i], \quad (3.4)$$

3.2. Lokalisation

Die Punktwolken der um diese Translationen reduzierten Fixpunkte $\vec{R}_i := \vec{r}_i - \vec{r}_0$ und $\vec{S}_i := \vec{s}_i - \vec{s}_0$ werden mit der Drehstreckung (m, φ) optimal ineinander eingepasst:

$$m = \frac{\sqrt{\left(\sum_{i=1}^n [R_{ix} S_{iy} + R_{iy} S_{ix}]\right)^2 + \left(\sum_{i=1}^n [R_{iy} S_{ix} - R_{ix} S_{iy}]\right)^2}}{\sum_{i=1}^n [S_{ix}^2 + S_{iy}^2]}, \quad (3.5)$$

$$\varphi = \arg \left(\left(\frac{\sum_{i=1}^n [R_{ix} S_{iy} + R_{iy} S_{ix}]}{\sum_{i=1}^n [R_{iy} S_{ix} - R_{ix} S_{iy}]} \right) \right). \quad (3.6)$$

Dabei wird mit dem quadrantenbereinigtem Arkustangens

$$\vec{r} = \begin{pmatrix} r_x \\ r_y \end{pmatrix} \mapsto \arg(\vec{r}) = \begin{cases} \arg : \mathbb{R}^2 \rightarrow \mathbb{R} \\ \left\{ \begin{array}{ll} \arctan\left(\frac{r_y}{r_x}\right) \operatorname{sign}(r_y) & , r_x > 0 \\ \frac{1}{2}\pi \operatorname{sign}(r_y) & , r_x = 0 \\ \left(\pi - \arctan\left(\frac{r_y}{r_x}\right)\right) \operatorname{sign}(r_y) & , r_x < 0 \end{array} \right. & , r_y \neq 0 \\ \left\{ \begin{array}{ll} 0 & , r_x > 0 \\ \text{nicht definiert} & , r_x = 0 \\ \pi & , r_x < 0 \end{array} \right. & , r_y = 0 \end{cases} \quad (3.7)$$

die Orientierung eines Vektors bezüglich \vec{E}_1 bestimmt. Wird nach Bestimmung der optimalen Transformationsparameter anhand 3.3 bis 3.6 der Ursprung des Beobachtersystems durch Gleichung 3.1 in das Referenzsystem abgebildet, ist die Aufgabe der Positionsbestimmung gelöst.

3.2.2. Helmert-Transformation mit Wichtungen

Aus den mit Hilfe der Bildverarbeitung bestimmten Segmenten als Mengen untereinander verbundener Bildelemente gleicher Farbe lassen sich sowohl Schwerpunkte, als auch Flächeninhalte bestimmen. Diese Eigenschaften der Segmente bzw. der ihnen zugrunde liegenden Markierungen sind die Eingabedaten für die Lagebestimmung. Bei der Lagebestimmung mit Hilfe der Helmert-Transformation nach [GJ07] werden nur die Segmentschwerpunkte berücksichtigt, die Zusatzinformationen der Segmentflächeninhalte bleiben ungenutzt.

3. Symbolische Auswertung – Bildinterpretation

Hieraus ergibt sich der Wunsch, aus den zusätzlichen Informationen über detektierte und erwartete Flächeninhalte Gütekriterien abzuleiten, welche als Wichtungen ω_i der korrespondierenden Einzelfehler $\vec{\varepsilon}_i$ benutzt werden. Somit soll erreicht werden, dass stark gestörte bzw. schlecht erkannte Segmente entsprechend wenig Einfluss auf das Ergebnis haben.

Der zu minimierende Gesamtfehler ergibt sich nach (3.1) somit zu:

$$\begin{aligned}
 E &= \sum_{i=1}^n [(\omega_i \vec{\varepsilon}_i)^2] \\
 &= \sum_{i=1}^n \left[\omega_i^2 \left(m \cos(\varphi) (s_{i_x} - s_{0_x}) - m \sin(\varphi) (s_{i_y} - s_{0_y}) - (r_{i_x} - r_{0_x}) \right)^2 \right] \\
 &+ \sum_{i=1}^n \left[\omega_i^2 \left(m \sin(\varphi) (s_{i_x} - s_{0_x}) + m \cos(\varphi) (s_{i_y} - s_{0_y}) - (r_{i_y} - r_{0_y}) \right)^2 \right].
 \end{aligned} \tag{3.8}$$

Notwendige Bedingungen für ein Minimum des Gesamtfehlers sind

$$\begin{aligned}
 \frac{\partial E}{\partial r_{0_x}} &= \\
 2 \sum_{i=1}^n \left[m \omega_i^2 \left(\cos(\varphi) (s_{i_x} - s_{0_x}) - \sin(\varphi) (s_{i_y} - s_{0_y}) - \left(\frac{r_{i_x}}{m} - \frac{r_{0_x}}{m} \right) \right) \right] &\stackrel{!}{=} 0
 \end{aligned} \tag{3.9}$$

und

$$\begin{aligned}
 \frac{\partial E}{\partial r_{0_y}} &= \\
 2 \sum_{i=1}^n \left[m \omega_i^2 \left(\sin(\varphi) (s_{i_x} - s_{0_x}) + \cos(\varphi) (s_{i_y} - s_{0_y}) - \left(\frac{r_{i_y}}{m} - \frac{r_{0_y}}{m} \right) \right) \right] &\stackrel{!}{=} 0.
 \end{aligned} \tag{3.10}$$

Aus (3.9) folgt

$$\begin{aligned}
 & m \cos(\varphi) \sum_{i=1}^n [\omega_i^2 s_{ix}] - m \sin(\varphi) \sum_{i=1}^n [\omega_i^2 s_{iy}] - \sum_{i=1}^n [\omega_i^2 r_{ix}] \\
 &= m \cos(\varphi) s_{0x} \left[\sum_{i=1}^n \omega_i^2 \right] - m \sin(\varphi) s_{0y} \left[\sum_{i=1}^n \omega_i^2 \right] - r_{0x} \sum_{i=1}^n [\omega_i^2] \quad (3.11)
 \end{aligned}$$

und ein Koeffizientenvergleich ergibt:

$$\cos(\varphi) : m \sum_{i=1}^n [\omega_i^2 s_{ix}] = m s_{0x} \sum_{i=1}^n [\omega_i^2] \quad (3.12)$$

$$\sin(\varphi) : -m \sum_{i=1}^n [\omega_i^2 s_{iy}] = -m s_{0y} \sum_{i=1}^n [\omega_i^2] \quad (3.13)$$

$$\text{Absolutglied} : -\sum_{i=1}^n [\omega_i^2 r_{ix}] = -r_{0x} \sum_{i=1}^n [\omega_i^2] . \quad (3.14)$$

Analog folgt aus (3.10)

$$\begin{aligned}
 & m \sin(\varphi) \sum_{i=1}^n [\omega_i^2 s_{ix}] + m \cos(\varphi) \sum_{i=1}^n [\omega_i^2 s_{iy}] - \sum_{i=1}^n [\omega_i^2 r_{iy}] \\
 &= m \sin(\varphi) s_{0x} \sum_{i=1}^n [\omega_i^2] + m \cos(\varphi) s_{0y} \sum_{i=1}^n [\omega_i^2] - r_{0y} \sum_{i=1}^n [\omega_i^2] \quad (3.15)
 \end{aligned}$$

und somit

$$\sin(\varphi) : m \sum_{i=1}^n [\omega_i^2 s_{ix}] = m s_{0x} \sum_{i=1}^n [\omega_i^2] \quad (3.16)$$

$$\cos(\varphi) : m \sum_{i=1}^n [\omega_i^2 s_{iy}] = m s_{0y} \sum_{i=1}^n [\omega_i^2] \quad (3.17)$$

$$\text{Absolutglied} : -\sum_{i=1}^n [\omega_i^2 r_{iy}] = -r_{0y} \sum_{i=1}^n [\omega_i^2] . \quad (3.18)$$

3. Symbolische Auswertung – Bildinterpretation

Hiermit ergeben sich die Translationen zu

$$\vec{s}_0 = \frac{\sum_{i=1}^n [\omega_i^2 \vec{s}_i]}{\sum_{i=1}^n [\omega_i^2]} \quad (3.19)$$

und

$$\vec{r}_0 = \frac{\sum_{i=1}^n [\omega_i^2 \vec{r}_i]}{\sum_{i=1}^n [\omega_i^2]} \quad (3.20)$$

als gewichtete Schwerpunkte der Fixpunkte in den jeweiligen Systemen.

Substitution der Terme $\vec{R}_i := \vec{r}_i - \vec{r}_0$ und $\vec{S}_i := \vec{s}_i - \vec{s}_0$ sowie $a := m \cos(\varphi)$ und $o := m \sin(\varphi)$ bzw. $\vec{E}_1 = \begin{pmatrix} a & o \end{pmatrix}^T$ führt zu den partiellen Ableitungen

$$\frac{\partial E}{\partial a} = 2 \sum_{i=1}^n \left[\omega_i^2 \left((aS_{ix} - oS_{iy} - R_{ix})S_{ix} + (oS_{ix} + aS_{iy} - R_{iy})S_{iy} \right) \right] \stackrel{!}{=} 0 \quad (3.21)$$

und

$$\frac{\partial E}{\partial o} = 2 \sum_{i=1}^n \left[\omega_i^2 \left((-aS_{ix} + oS_{iy} + R_{ix})S_{iy} + (oS_{ix} + aS_{iy} - R_{iy})S_{ix} \right) \right] \stackrel{!}{=} 0. \quad (3.22)$$

Gleichung (3.21) eignet sich zur Bestimmung des Parameters a :

$$\begin{aligned}
 \sum_{i=1}^n [\omega_i^2 (R_{i_x} S_{i_x} + R_{i_y} S_{i_y})] \\
 &= \sum_{i=1}^n [\omega_i^2 ((aS_{i_x} - oS_{i_y})S_{i_x} + (oS_{i_x} + aS_{i_y})S_{i_y})] \\
 &= a \sum_{i=1}^n [\omega_i^2 (S_{i_x}^2 + S_{i_y}^2)], \quad (3.23)
 \end{aligned}$$

analog kann aus (3.22) der Parameter o bestimmt werden:

$$\begin{aligned}
 \sum_{i=1}^n [\omega_i^2 (-R_{i_x} S_{i_y} + R_{i_y} S_{i_x})] \\
 &= \sum_{i=1}^n [\omega_i^2 ((-aS_{i_x} + oS_{i_y})S_{i_y} + (oS_{i_x} + aS_{i_y})S_{i_x})] \\
 &= o \sum_{i=1}^n [\omega_i^2 (S_{i_x}^2 + S_{i_y}^2)]. \quad (3.24)
 \end{aligned}$$

Aus $a^2 + o^2 = m^2(\cos^2(\varphi) + \sin^2(\varphi)) = m^2$ ergibt sich

$$\begin{aligned}
 m &= \sqrt{\left(\frac{\sum_{i=1}^n [\omega_i^2 (R_{i_x} S_{i_x} + R_{i_y} S_{i_y})]}{\sum_{i=1}^n [\omega_i^2 (S_{i_x}^2 + S_{i_y}^2)]} \right)^2 + \left(\frac{\sum_{i=1}^n [\omega_i^2 (R_{i_y} S_{i_x} - R_{i_x} S_{i_y})]}{\sum_{i=1}^n [\omega_i^2 (S_{i_x}^2 + S_{i_y}^2)]} \right)^2} \\
 &= \sqrt{\left(\sum_{i=1}^n [\omega_i^2 (R_{i_x} S_{i_x} + R_{i_y} S_{i_y})] \right)^2 + \left(\sum_{i=1}^n [\omega_i^2 (R_{i_y} S_{i_x} - R_{i_x} S_{i_y})] \right)^2} \\
 &\quad \frac{1}{\sum_{i=1}^n [\omega_i^2 (S_{i_x}^2 + S_{i_y}^2)]}. \quad (3.25)
 \end{aligned}$$

3. Symbolische Auswertung – Bildinterpretation

Entsprechend der Substitutionen für a und o gilt $\frac{a}{m} = \cos(\varphi)$ und $\frac{o}{m} = \sin(\varphi)$. Dies ist erfüllt durch

$$\varphi = \arg \left(\frac{\left(\frac{\sum_{i=1}^n [\omega_i^2 (R_{ix} S_{ix} + R_{iy} S_{iy})]}{\sum_{i=1}^n [\omega_i^2 (S_{ix}^2 + S_{iy}^2)]} \right)}{\left(\frac{\sum_{i=1}^n [\omega_i^2 (R_{iy} S_{ix} - R_{ix} S_{iy})]}{\sum_{i=1}^n [\omega_i^2 (S_{ix}^2 + S_{iy}^2)]} \right)} \right) = \arg \left(\frac{\left(\sum_{i=1}^n [\omega_i^2 (R_{ix} S_{ix} + R_{iy} S_{iy})]}{\sum_{i=1}^n [\omega_i^2 (R_{iy} S_{ix} - R_{ix} S_{iy})]} \right)}{\left(\sum_{i=1}^n [\omega_i^2 (R_{iy} S_{ix} - R_{ix} S_{iy})]} \right)} \right). \quad (3.26)$$

Wieder kann die Position des Spielers durch Transformation des Spielerurprungs nach (3.1) ins Referenzsystem bestimmt werden, dabei gehen die Transformationsparameter nach (3.19), (3.20), (3.25) und (3.26) für Wichtungen $w_i = 1$ in die Parameter nach (3.3), (3.4), (3.5) und (3.6) über.

3.2.3. Vergleich verschiedener Gütekriterien

Zur Überprüfung der Vermutung, dass der Einsatz von Wichtungen ω_i zur Verbesserung der Genauigkeit der Lagebestimmung führen kann, wird ein synthetischer Test durchgeführt. Mit Hilfe eines vereinfachten Trikotmodells werden erwartete Schwerpunktkoordinaten und -flächeninhalte generiert und als exakte Daten des Beobachtersystems benutzt. Mit einem Störmodell werden die Trikotparameter modifiziert und hierdurch konsistent degenerierte Messwerte simuliert. Diese werden als Fixpunktkoordinaten des Referenzsystems interpretiert. Aus den so erzeugten Testdaten werden die Transformationsparameter nach (3.19), (3.20), (3.25) und (3.26) bestimmt. Aus vorgegebener und berechneter Spielerlage ergibt sich die resultierende Abweichung der Lagebestimmung.

Vereinfachtes Markierungsmodell

Zur Abschätzung der Einflüsse von Klassifizierungsfehlern auf die Genauigkeit der Lagebestimmung sei das Markierungsmodell folgendermaßen definiert: Es werden die Markierungen A , B , C und D jeweils als Kreisringsektoren der Parameter Mittelpunkt (x_0, y_0) , Innenradius r_i , Außenradius r_a , Startwinkel

φ_0 und Endwinkel φ_1 betrachtet. Diese Markierungsparameter finden sich in Tabelle 3.1.

Markierung	x_0	y_0	r_i	r_a	φ_0	φ_1
A	0	0	0	25mm	0	2π
B	0	0	25mm	90mm	$\pi/6$	$\varphi_0 + 2\pi/3$
C	0	0	25mm	90mm	$5\pi/6$	$\varphi_0 + 2\pi/3$
D	0	0	25mm	90mm	$3\pi/2$	$\varphi_0 + 2\pi/3$

Tabelle 3.1.: Parameter des vereinfachten Markierungsmodells

Die Einzelfläche A eines solchen Kreisringsektors ist

$$A = \int_{r_i}^{r_a} \int_{\varphi_0}^{\varphi_1} r d\varphi dr = \frac{1}{2}(\varphi_1 - \varphi_0)(r_a^2 - r_i^2). \quad (3.27)$$

der Flächenschwerpunkt \vec{s} folgt

$$\begin{aligned} \vec{s} &= \frac{1}{A} \int_{r_i}^{r_a} \int_{\varphi_0}^{\varphi_1} \begin{pmatrix} x_0 + r \cos(\varphi) \\ y_0 + r \sin(\varphi) \end{pmatrix} r d\varphi dr \\ &= \frac{1}{A} \left(\frac{r_a^3 - r_i^3}{3} \begin{pmatrix} \sin(\varphi_1) - \sin(\varphi_0) \\ -\cos(\varphi_1) + \cos(\varphi_0) \end{pmatrix} + \frac{r_a^2 - r_i^2}{2} (\varphi_1 - \varphi_0) \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \right). \end{aligned} \quad (3.28)$$

Trikotstörmodell

Zur Simulation der Einflüsse der Fehlklassifikation bei der Bildverarbeitung auf die Genauigkeit der Lagebestimmung wird ein Trikot durch Manipulation der Parameter seiner Kreisringsektoren entsprechend Algorithmus 7 und Algorithmus 8 degradiert. Hierdurch wird erreicht, dass die resultierenden simulierten Störungen konsistent sind, also tatsächlich als Ergebnis der Bildverarbeitung auftreten können. Bei einem alternativen Vorgehen, der direkten Beeinflussung von Schwerpunkten und -flächen, ist dies nicht sichergestellt. Abbildung 3.4 verdeutlicht die Anwendung der Algorithmen 7 und 8.

3. Symbolische Auswertung – Bildinterpretation

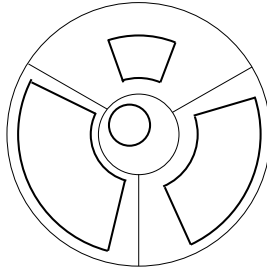


Abbildung 3.4.: Die Störungssimulation berücksichtigt nur die fehlerhafte Klassifikation einer Markierungsfarbe als Rest, die Fehlklassifikation zwischen den Markierungsfarben wird nicht modelliert.

Definition der Gütekriterien

Als Gütekriterien zur Bestimmung der Wichtung eines durch ein Segment verursachten Einzelfehlers werden folgende Varianten in Betracht gezogen:

Ungewichtet $w_i = 1$, Als Referenz werden alle Einzelfehler gleich gewichtet.

Fläche $w_i = A'_i$, Eine große gemessene Fläche lässt vergleichsweise kleine Auswirkungen stochastischer Effekte vermuten. Die Fläche des Segmentes im Referenzsystem wird zur Wichtung eines Einzelfehlers genutzt.

Flächenverhältnis $w_i = \frac{A'_i}{A_i}$, Das Verhältnis aus gemessener Segmentfläche (Referenzsystem) und erwarteter Fläche (Beobachtersystem) zeigt an, wie gut das Segment erkannt wurde und dient als Wichtung des Einzelfehlers.

Bestimmung der Lageabweichung für eine einzelne Störung

Anhand des vereinfachten Trikotmodells und des Trikotstörmodells ist die Auswirkung simulierter Fehlklassifizierungen auf die Genauigkeit der Lagebestimmung direkt bestimmbar:

1. Lege die Roboterlage zu $\begin{pmatrix} x & y & \varphi \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$ fest, die Trikotparameter im Beobachtersystem und im Referenzsystem sind somit gleich.
2. Verändere die Trikotparameter für das Referenzsystem entsprechend dem Trikotstörmodell.
3. Bestimme anhand des Trikotmodells die Flächen und Flächenschwerpunkte für Beobachtersystem und Referenzsystem.

Algorithmus 7: Störung der Zentralmarke

```

1 Beginn
   /*Der Flächenstörgrad bzw. die Segmentgüte gibt an, wie viel
   Fläche für das gestörte Segment zur Verfügung stehen soll. */
2   Lege eine Segmentgüte  $\frac{A'}{A} \in (0, 1]$  fest.
   /*Dieser Segmentgüte entspricht der Radius des
   Zentralsegmentes */
3   Berechne zugehörigen Radius  $r'_a = r_a \sqrt{\frac{A'}{A}}$ .
   /*Das gestörte Zentralsegment soll innerhalb des ungestörten
   Zentralsegmentes liegen, hieraus ergeben sich die
   Koordinaten des Ursprungs mit  $x_{\max} = r_a - r'_a$ . */
4   Lege  $x'_0 \in [-x_{\max}, x_{\max}]$  fest.
   /*und über den Satz des Pythagoras folgt  $y_{\max} = \sqrt{x_{\max}^2 - x'^2_0}$ . */
5   Lege  $y'_0 \in [-y_{\max}, y_{\max}]$  fest.

```

4. Bestimme die Lage $(x' \ y' \ \varphi')^T$ aus den Flächen und Flächenschwerpunkten in Beobachtersystem und Referenzsystem für das betrachtete Gütekriterium.
5. Die Differenz $(\Delta x \ \Delta y \ \Delta \varphi)^T = (x' \ y' \ \varphi')^T - (x \ y \ \varphi)^T$ aus ermittelter und vorgegebener Lage ist die durch die Störung verursachte Lageabweichung.

Wertung einer Lageabweichung

Der Vergleich der verschiedenen Gütekriterien bei der Lagebestimmung soll am Versagensfall geschehen. Versagen der Lagebestimmung wird angenommen, wenn die Lageabweichung so groß ist, dass ein Schuss nicht sicher ausgeführt werden kann. Hierfür werden folgende Kriterien betrachtet:

1. Aus der Breite der Kontaktfläche des Schussmechanismus ergibt sich eine Toleranz von ± 45 mm für die x -Richtung.
2. Aus dem Maximalhub des Schussmechanismus resultiert die Toleranz in y -Richtung zu $\pm 7,5$ mm.
3. Die Wichtung des Fehlwinkels kann anhand einer distanzbezogenen Schusstoleranz bestimmt werden: Auf eine Entfernung von 3 m werden 0,2 m Abweichung toleriert.

3. Symbolische Auswertung – Bildinterpretation

Algorithmus 8: Störung einer Außenmarke

```

1 Beginn
2   Lege Segmentgüte  $\frac{A'}{A} \in (0, 1]$  fest.
   /*Der überstrichene Winkel  $\Delta\varphi = \varphi_1 - \varphi_0$  des ungestörten
     Segmentes dient zur Bestimmung des Wertebereiches für den
     überstrichenen Winkel  $\Delta\varphi' = \varphi'_1 - \varphi'_0$  des gestörten Segmentes.
   */
3   Lege  $\Delta\varphi' \in [\frac{A'}{A} \Delta\varphi, \Delta\varphi]$  fest.
   /*Mit dem überstrichenen Winkel  $\Delta\varphi'$  ergibt sich der minimale
     Außenradius des Segmentes  $r_{\min} = \sqrt{2 \frac{A'}{\Delta\varphi'}} + r_i^2$ , für den die
     Segmentgüte  $\frac{A'}{A}$  erreicht werden kann.                                     */
4   Lege Außenradius  $r'_a \in [r_{\min}, r_a]$ .
5   Berechne den dazu passenden Innenradius  $r'_i = \sqrt{r_a'^2 - 2 \frac{A'}{\Delta\varphi'}}$ 
   /*Durch das Verteilen der „Lücke“  $\Delta\varphi - \Delta\varphi'$  mit einem
     „Lückenparameter“  $z \in [0, 1]$  ergibt sich der Startwinkel  $\varphi'_0$ . */
6   Lege  $\varphi'_0 = \varphi_0 + z(\Delta\varphi - \Delta\varphi')$  fest.

```

Aus diesen Kriterien wird der normierte Fehler δ zu

$$\delta = \sqrt{\left(\frac{\Delta x}{45 \text{ mm}}\right)^2 + \left(\frac{\Delta y}{7,5 \text{ mm}}\right)^2 + \left(\frac{\Delta\varphi}{\arctan\left(\frac{0,2 \text{ m}}{3 \text{ m}}\right)}\right)^2} \quad (3.29)$$

aufgestellt, wobei $\delta > 1$ entsprechend der Kriterien Versagen signalisiert.

Ergebnisse

Das Experiment soll die Frage beantworten, ob das Ergebnis der optimalen Lagebestimmung nach Helmert durch Einsatz von Zusatzwissen verbessert werden kann. Hierzu wird der Einfluss der verschiedenen Gütekriterien auf die Genauigkeit der Lagebestimmung empirisch anhand eines synthetischen Tests untersucht. Dabei wird das Trikotmodell zur Bestimmung der Flächeninformationen genutzt. Über das Trikotstörmodell werden konsistente Störungen generiert. Die Störung eines Robotermodells äußert sich neben der Verschiebung der jeweiligen Segmentenschwerpunkte in einer Gesamtgüte $A'/A = \sum_i A'_i / \sum_i A_i$. Diese Güte entspricht der mittleren simulierten Fehlklassifikationsrate des Trikots. Für jede Gesamtgüte existiert dabei ein Bereich möglicher realisierter normierter Fehler.

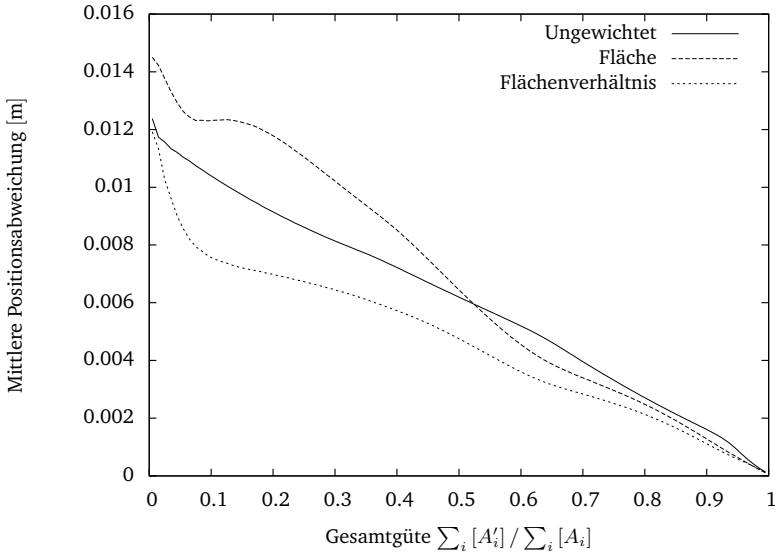


Abbildung 3.5.: Mittlere Positionsabweichungen

Im Experiment werden 10^9 Trikotstörungen durch die Algorithmen 7 und 8 erzeugt. Für jede Störung wird die zugehörige Gesamtgüte und die resultierende Lageabweichung bestimmt. Hieraus werden die Positionsabweichungen, die Winkelabweichungen und die normierten Lageabweichungen über 100 Gesamtgüteklassen aggregiert und die entsprechenden Mittelwerte über den Güteklassen bestimmt. Diese Mittelwerte werden in den Abbildungen 3.5-3.7 gegen die Güteklassen aufgetragen. In Diagramm 3.7 kann das Versagen beim Einsatz der verschiedenen Gütekriterien verglichen werden. Unterhalb einer bestimmten Gesamtgütekategorie signalisiert das Klassenmittel der normierten Lageabweichungen Versagen. Die Grenzgüte sei der Mittelwert der Grenzen dieser Gesamtgütekategorie. Tabelle 3.2 liefert eine Übersicht der ermittelten Grenzgüten. Das Experiment zeigt, dass die Gütekriterien Fläche und Flächenverhältnis sich in der Grenzgüte kaum unterscheiden. Bei beiden Varianten ist die Grenzgüte kleiner als beim ungewichteten Verfahren. Dies kann als Gewinn an Fehlklassifikationstoleranz bzw. Robustheit der Lagebestimmung verstanden werden. Diese Steigerung der Robustheit resultiert aus der Nutzung von Zusatzwissen durch den Einsatz von Einzelfehlerwichtungen in der Helmert-Transformation.

3. Symbolische Auswertung – Bildinterpretation

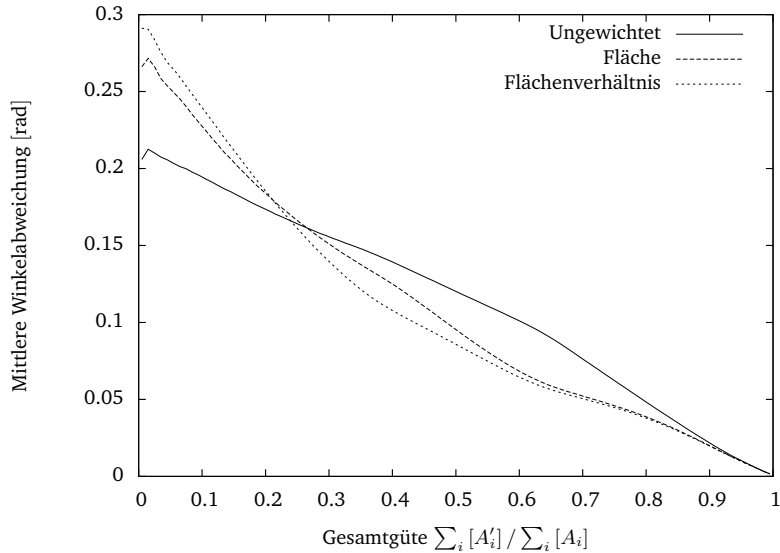


Abbildung 3.6.: Mittlere Winkelabweichungen

Gütekriterium	Grenzgüte
Ungewichtet	0,69
Fläche	0,57
Flächenverhältnis	0,54

Tabelle 3.2.: Grenzgüten beim Einsatz der verschiedenen Gütekriterien

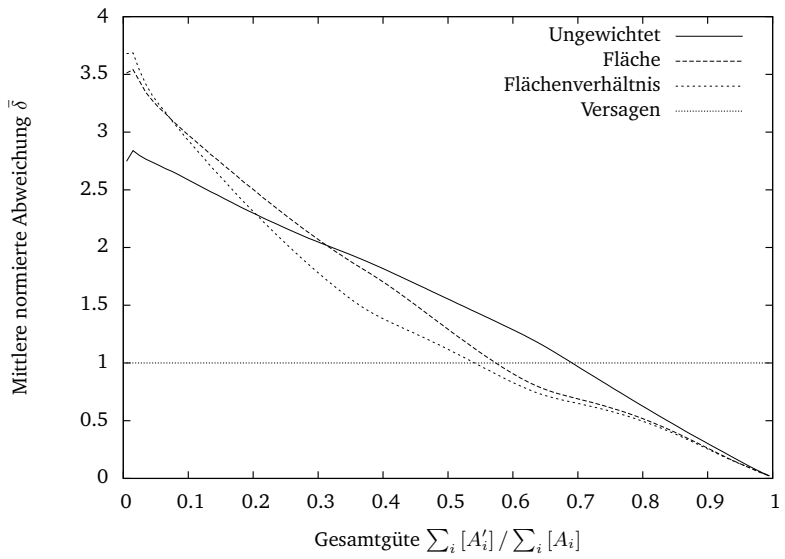


Abbildung 3.7.: Mittlere normierte Lageabweichungen

A monk said to Tozan
*A monk has died; where has
he gone?*
Tozan answered
After the fire, a sprout of grass.

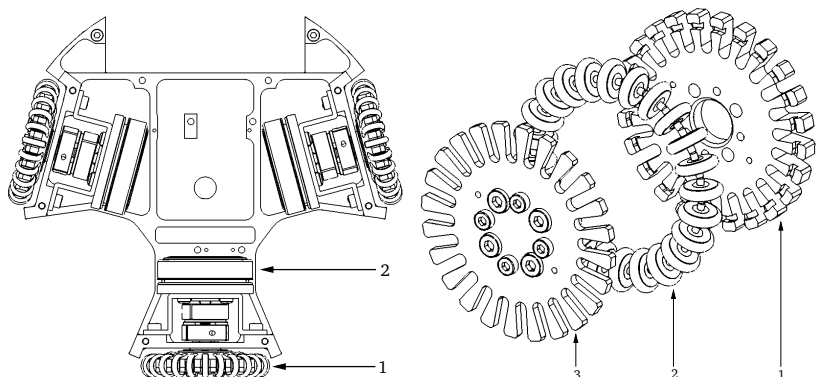
(Zen Kōan)

Teil II.

Steuerung eines RoboCup-Spielers

4. Regelung eines omnidirektionalen Antriebssystems

Eine wichtige Aufgabe beim autonomen Fußballspiel ist die Überführung eines Roboters aus seiner aktuellen Lage in eine Ziellage. Die verwendeten Roboter verfügen über omnidirektionale Antriebssysteme, die eine beliebige Bewegung in Längs- und Querrichtung sowie Rotation erlauben, siehe Abbildung 4.1.



(a) Omnidirektionalrad (1) und Elektromotor (2) (b) Omnidirektionalrad bestehend aus Innenschale (1), Kranzrädern (2) und Außenschale (3)

Abbildung 4.1.: Omnidirektionales Antriebssystem

Für die Regelung eines solchen Antriebssystems können durch die Robotergeometrie die Winkelgeschwindigkeiten der einzelnen Räder für eine angestrebte Robotergeschwindigkeit bestimmt werden. Im einfachsten Fall werden diese durch unterlagerte und voneinander unabhängige Winkelgeschwindigkeitsregelkreise für die Räder (Antriebsregelkreise) beeinflusst. Eine solche Struktur der kinematischen Steuerung mit unterlagerten Antriebsregelkreisen ist in den Abbildungen 4.2 und 4.3 dargestellt.

4. Regelung eines omnidirektionalen Antriebssystems

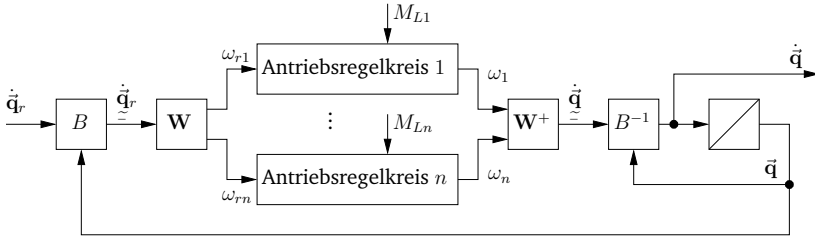


Abbildung 4.2.: Geschwindigkeitssteuerung mit unterlagerten Antriebesregelkreisen: Die Sollgeschwindigkeit $\dot{\vec{q}}_r$ im globalen System (siehe Abbildung 4.5(a)) wird in das Roboterkoordinatensystem (siehe Abbildung 4.5(b)) projiziert. Die Matrix W (siehe (4.89)) modelliert die Verknüpfung aus Robotergeschwindigkeit und Radwinkelgeschwindigkeiten anhand der Rollbedingung. Durch diese Verknüpfung ergeben sich die Führungsgrößen ω_{ri} der unabhängigen Antriebsregelkreise. Die Radwinkelgeschwindigkeiten ω_i , können nach [RF06] in die ins Roboterkoordinatensystem projizierte Geschwindigkeit überführt werden. Hiermit wird durch Projektion in das ortsfeste Koordinatensystem die Robotergeschwindigkeit $\dot{\vec{q}}$ bestimmt. Integration der Robotergeschwindigkeit über die Zeit führt zur Roboterlage $\vec{q} = (r_x \ r_y \ \Psi)^T$, wobei die Ausrichtung Ψ des Roboters im ortsfesten System die Drehungen durch B und B^{-1} festlegt. Die durch die mechanische Verkopplung des Systems bedingten Lastmomente M_{Li} bleiben dabei unberücksichtigt, treten jedoch als Störungen der Antriebsregelkreise zutage.

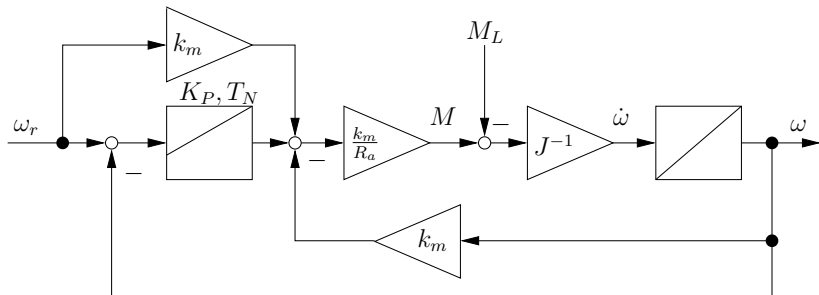


Abbildung 4.3.: Vereinfachter Drehzahlregelkreis für einen Antrieb: Die Differenz aus Sollwinkelgeschwindigkeit ω_r und Radwinkelgeschwindigkeit ω ist Eingangsgröße eines PI-Reglers (vgl. [Föl92, Lun08]). Dessen Ausgang ist die Klemmspannung des Antriebs. Die um die Induktionsspannung $k_m\omega$ verminderte Klemmspannung treibt den Strom durch den Ankerstromkreis des Widerstandes R_a und erzeugt bei der Maschinenkonstante k_m das Antriebsmoment M . Die Differenz aus Antriebsmoment und Lastmoment M_L führt am Antriebsstrang der Trägheit J zur Winkelbeschleunigung $\dot{\omega}$ bzw. nach Integration zur Winkelgeschwindigkeit ω . Mittels Vorsteuerung wird der Effekt der Induktionsspannung stationär kompensiert. Eine mögliche Übersetzung zwischen Antrieb und Rad, die Zeitkonstante des Ankerstromkreises sowie die Begrenzung der Klemmspannung sind in diesem Beispiel nicht berücksichtigt.

4. Regelung eines omnidirektionalen Antriebssystems

Bei diesem Ansatz ergibt sich das Problem der Verkopplung der Antriebsregelkreise durch die Lastmomente, welche aus der Beschleunigung des Systems resultieren. Die Verkopplung wirkt hierbei als Störung und bewirkt beschleunigungsabhängige Geschwindigkeitsfehler. Dadurch wird die Einsatzfähigkeit der Struktur erheblich beeinträchtigt.

Die Verkopplung des mechanischen Systems durch die Lastmomente ist ein systematischer Effekt und kann durch inverse Dynamikmodelle kompensiert werden [CGD⁺01, Lan05, RF06, LZWW08]. Für diese Kompensation wird eine Erweiterung der Struktur um eine Vorsteuerung der Antriebsregelkreise beschrieben [CGD⁺01, LZWW08], bei der die Begrenzung der Beschleunigungsfähigkeit durch die Haftreibungsgrenze unberücksichtigt bleibt. Hieraus resultierendes Durchdrehen eines Rades wird dabei durch redundante Antriebssysteme detektiert und zur Korrektur der Führungsgröße der kinematischen Steuerung verwendet.

In der vorliegenden Arbeit wird die Beschränkung der Beschleunigungsfähigkeit eines omnidirektionalen Antriebssystems durch elektrische und mechanische Nebenbedingungen mit Hilfe eines Mehrkörpermodells beschrieben. Hierbei werden sowohl die Skalierung einer vorgegebenen Beschleunigung auf das realisierbare Maximum als auch die entsprechenden Antriebskräfte bestimmt. Dies geschieht durch eine Formulierung als lineares Optimierungsproblem, welches mit Hilfe des Simplex-Verfahrens [Dan66, GRHHM89] gelöst wird. Hiermit wird das Modell für den Einsatz in einem Regler anwendbar.

Somit baut sich dieses Kapitel wie folgt auf: In Abschnitt 4.1 wird ein Mehrkörpermodell bestehend aus Grundkörper, Rädern und Antrieben zur Beschreibung des Zusammenhanges zwischen Roboterbeschleunigung sowie Antriebskräften und -momenten dargelegt. Wichtige Zwischenschritte der Ausarbeitung dieses Modells werden mit Hilfe des Computeralgebrasystems „Maxima“ [max09] durchgeführt. Im Anhang A findet sich ein Beispiel für die Verwendung¹ von Maxima zur Formulierung des Modells.

In Abschnitt 4.2 folgt die Aufbereitung dieses Modells zur numerischen Auswertung in einem Regler in Anlehnung an [RF06]. Im Gegensatz zur dort fokussierten energieoptimalen Ansteuerung und dem Ausnutzen der Antriebsredundanz zur Schlupfdetektion wird hier der Schwerpunkt auf mechanische und elektrische Begrenzungen der Beschleunigungsfähigkeit gelegt. Ein alternatives Modell eines omnidirektionalen Roboters ist in [CGD⁺01] beschrieben, jedoch verzichtet dieses auf die Beschreibung der Begrenzungen. In [WCGR02] wird ein nichtlineares Dynamikmodell für omnidirektionale Roboter mit Radschlupf dargelegt. Aspekte dieses Modells werden in vereinfachter Form aufgegriffen.

¹Maxima wurde als Hilfsmittel zur Durchführung von Koordinatentransformationen, zur Bestimmung zeitlicher Ableitungen und zur Ersetzung von Ausdrücken benutzt.

4.1. Modellierung eines omnidirektional getriebenen Roboters

Die ausführliche Modellbildung wird durch mehrere Aspekte motiviert. Das bereitgestellte Modell ermöglicht Parameteruntersuchungen und kann so die technische Auslegung zukünftiger Spieler unterstützen. Es eröffnet die Möglichkeit, optimale Regelungsstrategien [Föl88] bzw. Approximationen für Echtzeitanwendungen [Büs11] zu entwerfen. Das Modell ermöglicht den Einsatz von Zustandsbeobachtern [Kal60, CC99, TBF05, Ewa05], um durch die Fusion von Messwerten mit den Vorhersagen des Modells die Robustheit des Systems gegenüber Messrauschen bzw. gegenüber Totzeiten [LARP98, BEG⁺03] zu erhöhen.

Im Abschnitt 4.3 werden ein Geschwindigkeits- und zwei Lageregler erstellt. Diese Regler basieren auf der Auswertung des Modells. Die Regler finden Anwendung im Modul der simulierten Umgebung.

4.1. Modellierung eines omnidirektional getriebenen Roboters als Mehrkörpersystem

Der Geschwindigkeitsregler in Abschnitt 4.3.1 soll auftretende Geschwindigkeitsfehler mit maximal möglicher Beschleunigung ausregeln. Hierfür wird die zustandsabhängige Beschleunigungsgrenze anhand eines Mehrkörpermodells ermittelt, siehe Abbildung 4.4. Grundlage dieses Modells sind Impuls- und Drehimpulssatz für die betrachteten Körper (Grundkörper, Antriebe und Räder). Die Kopplung der einzelnen Körper wird durch Schnittgrößen realisiert, siehe Abschnitt 4.1.3. Die Radanzahl des Modells ist größer zwei, wobei steuerbare Antriebskonfigurationen vorausgesetzt werden. Da das Modell ein Modul des Geschwindigkeitsreglers ist und somit unter der Forderung von Echtzeitfähigkeit in eingebetteten informationstechnischen Systemen ausgewertet werden muss, wird insbesondere der Abrollvorgang eines realen Allseitenrades nicht modelliert. Weiterhin findet der Antriebs- bzw. Bremschlupf in diesem Modell vorerst keine Berücksichtigung. Es wird jedoch eine Erweiterung des Modells um Radschlupf bereitgestellt. Bei der Bestimmung der maximalen Beschleunigung ergeben sich die benötigten Antriebskräfte als Teil der Lösung eines linearen Optimierungsproblems.

4. Regelung eines omnidirektionalen Antriebssystems

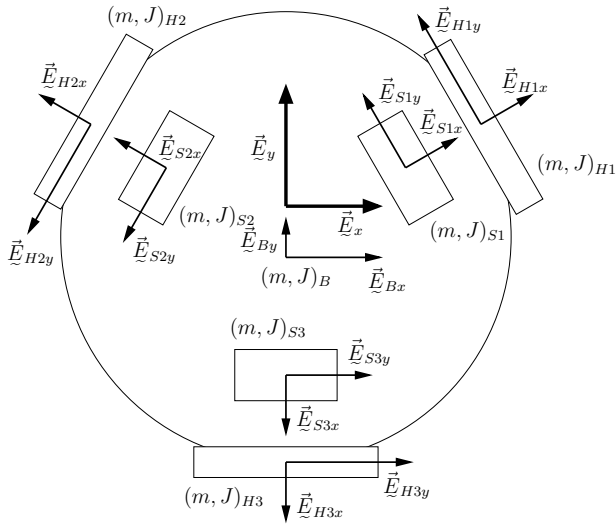
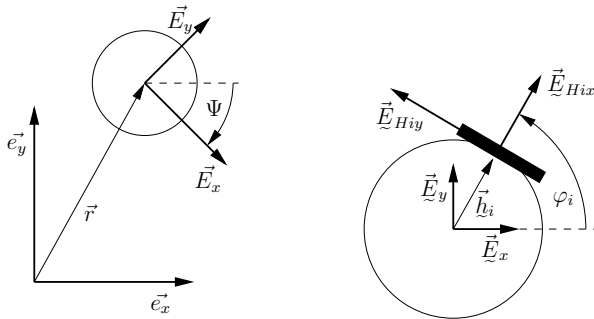


Abbildung 4.4.: Omnidirektionales Antriebssystem als Mehrkörpersystem



(a) Globales Koordinatensystem mit Roboter (Kreis) und eingeschriebenem Koordinatensystem in der Draufsicht

(b) Roboter (Kreis) mit lokalem Koordinatensystem sowie Rad (Balken) mit Nabenkoordinatensystem

4.1. Modellierung eines omnidirektional getriebenen Roboters

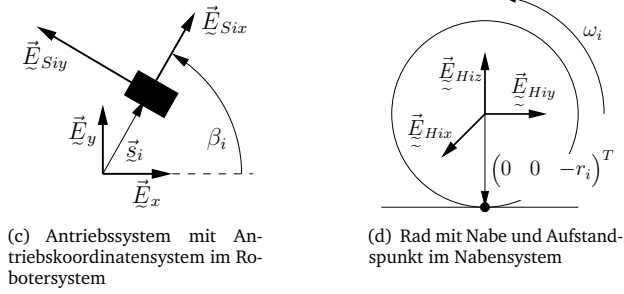


Abbildung 4.5.: Koordinatensysteme

4.1.1. Koordinatensysteme

Der Schwerpunkt $\vec{r} = (r_x \ r_y \ r_z)^T$ des Spielers im ortsfesten, globalem System sei der Ursprung des Robotersystems. Die Orientierung Ψ des Spielers führt zur Basis

$$B := \begin{pmatrix} \vec{E}_x^T \\ \vec{E}_y^T \\ \vec{E}_z^T \end{pmatrix} = \begin{pmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

des Robotersystems, wobei Ψ den Winkel zwischen den x -Achsen von globalem und Robotersystem bezeichnet, siehe Abbildung 4.5(a). Ein allgemeiner Wert v , der sich auf das Robotersystem bezieht, wird folgend durch \underline{v} ausgezeichnet. Die Schwerpunkte der modellierten Körper im Robotersystem sind $\underline{b} = (b_x \ b_y \ b_z)$ für den Grundkörper und $\underline{h}_i = (h_{ix} \ h_{iy} \ h_{iz})$ für das Rad i sowie $\underline{s}_i = (s_{ix} \ s_{iy} \ s_{iz})$ für den Antrieb i . Für jeden dieser Körper wird ein Körperkoordinatensystem im jeweiligen Schwerpunkt definiert, welches entlang der Hauptträgheitsachsen ausgerichtet ist. Ein allgemeiner Wert v , der sich auf ein Körperkoordinatensystem bezieht, wird folgend durch $\underline{\underline{v}}$ ausgezeichnet, wobei $\underline{\underline{v}}_B$ auf das Koordinatensystem des Grundkörpers, $\underline{\underline{v}}_{Hi}$ auf das Koordinatensystem des Rades i und $\underline{\underline{v}}_{Si}$ auf das Koordinatensystem des Antriebs i verweist. So bezeichnet beispielsweise $\underline{\underline{J}}_{Sixx}$ das Trägheitsmoment des Antriebs i um seine x -Achse.

4. Regelung eines omnidirektionalen Antriebssystems

Die Basis des Körperkoordinatensystems des Grundkörpers im Robotersystem (\vec{r}, B) sei

$$\underline{B}_B := \begin{pmatrix} \underline{\vec{E}}_{Bx}^T \\ \underline{\vec{E}}_{By}^T \\ \underline{\vec{E}}_{Bz}^T \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.2)$$

Die Basis des Körperkoordinatensystems des Rades i im Robotersystem sei

$$\underline{B}_{Hi} := \begin{pmatrix} \underline{\vec{E}}_{Hix}^T \\ \underline{\vec{E}}_{Hiy}^T \\ \underline{\vec{E}}_{Hiz}^T \end{pmatrix} = \begin{pmatrix} \cos(\varphi_i) & \sin(\varphi_i) & 0 \\ -\sin(\varphi_i) & \cos(\varphi_i) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (4.3)$$

wobei φ_i den Winkel zwischen den x -Achsen von Robotersystem und Radsystem bezeichnet, siehe Abbildung 4.5(b). Die x -Achse des Radsystems ist identisch der Radachse und die y -Achse liegt in der Radebene. Beide Achsen sind parallel zum Boden.

Analog hierzu sei die Basis der rotierenden Antriebsmasse i in (\vec{r}, B)

$$\underline{B}_{Si} := \begin{pmatrix} \cos(\beta_i) & \sin(\beta_i) & 0 \\ -\sin(\beta_i) & \cos(\beta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.4)$$

Dabei bezeichnet β_i den Winkel zwischen den x -Achsen von Robotersystem und Koordinatensystem des Antriebs, siehe Abbildung 4.5(c). Die x -Achse des Koordinatensystems der rotierenden Antriebsmasse ist identisch der Motorachse und die y -Achse liegt in der Rotorebene. Beide Achsen sind parallel zum Boden. Da die Basen aller Koordinatensysteme orthonormal sind, können die jeweiligen Inversen durch Transponieren gebildet werden.

4.1.2. Roboterzustand

Die Bewegung eines omnidirektionalen Roboters wird als planares Problem mit Freiheitsgrad 3 betrachtet. Beeinflussbare Größen der Roboterlage sind x - und y -Komponente des Roboterschwerpunktes sowie die Ausrichtung Ψ . Die *Roboterlage* wird im folgenden durch

$$\vec{q} = \begin{pmatrix} r_x & r_y & \Psi \end{pmatrix}^T \quad (4.5)$$

beschrieben. Die zeitliche Ableitung dieser Lage

$$\dot{\vec{q}} = \begin{pmatrix} \dot{r}_x & \dot{r}_y & \dot{\Psi} \end{pmatrix}^T \quad (4.6)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

wird im folgenden als *Roboter Geschwindigkeit* bezeichnet. Die zeitliche Ableitung dieser Roboter Geschwindigkeit

$$\ddot{\underline{\mathbf{q}}} = \begin{pmatrix} \ddot{r}_x & \ddot{r}_y & \ddot{\Psi} \end{pmatrix}^T \quad (4.7)$$

wird im folgenden als *Roboter beschleunigung* bezeichnet.

Der Zustand $\vec{q} \in \Omega$ des Roboters wird definiert zu

$$\vec{q} = \begin{pmatrix} \underline{\mathbf{q}}^T & \dot{\underline{\mathbf{q}}}^T \end{pmatrix}^T, \quad (4.8)$$

wobei Ω den Zustandsraum des Roboters bezeichnet.

Wichtige Vereinfachungen ergeben sich durch die lokale Beschreibung der Modellannahmen. Hierdurch wird ein großer Teil der Modellgleichungen zustandsinvariant, woraus Vorteile für die Implementierbarkeit des Modells unter Echtzeitanforderungen resultieren. Dabei bleibt das Modell im Rahmen der getroffenen Annahmen exakt. Für die lokale Beschreibung, der Beschreibung aus dem Robotersystem heraus, werden Projektionen der Roboter Geschwindigkeit

$$\dot{\underline{\mathbf{q}}} = \begin{pmatrix} \dot{\underline{r}}_x & \dot{\underline{r}}_y & \dot{\Psi} \end{pmatrix}^T = B^{-1} \ddot{\underline{\mathbf{q}}} \quad (4.9)$$

und der Roboter beschleunigung

$$\ddot{\underline{\mathbf{q}}} = \begin{pmatrix} \ddot{\underline{r}}_x & \ddot{\underline{r}}_y & \ddot{\Psi} \end{pmatrix}^T = B^{-1} \ddot{\underline{\mathbf{q}}} \quad (4.10)$$

genutzt. Weiterhin tritt im Modell ein gemischter quadratischer Geschwindigkeitsvektor

$$\dot{\underline{\mathbf{Q}}} = \begin{pmatrix} \dot{\underline{r}}_x \dot{\Psi} & \dot{\underline{r}}_y \dot{\Psi} & \dot{\Psi}^2 \end{pmatrix}^T \quad (4.11)$$

auf.

4.1.3. Schnittgrößen

Der Roboter wird als Mehrkörpersystem des Grundkörpers und der Räder sowie der rotierenden Teile der Antriebe modelliert. Der Grundkörper ist mit jedem Rad bzw. Antrieb verbunden, jedes Rad bzw. jeder Antrieb jeweils nur mit dem Grundkörper. Diese Kopplung wird mit den Schnittgrößen Schnittkräfte \vec{A}_{Hi} und Schnittmomente \vec{T}_{Hi} bezüglich der Räder sowie Schnittkräfte \vec{A}_{Si} und Schnittmomente \vec{T}_{Si} bezüglich der Antriebe realisiert. Die Schnittgrößen wirken jeweils positiv am Grundkörper und negativ am entsprechenden Rad bzw. Antrieb i .

4. Regelung eines omnidirektionalen Antriebssystems

4.1.4. Dynamik des Omnidirektionalrades

Geschwindigkeit und Winkelgeschwindigkeit des Omnidirektionalrades

Die Nabengeschwindigkeit des Omnidirektionalrades wird für das Aufstellen des Radimpulses bzw. des Impulssatzes für das Rad benötigt. Die Bestimmung seiner Winkelgeschwindigkeit um die Radachse ist einerseits notwendig, um seine Drehimpulsänderung als Grundlage des Drehimpulssatzes aufzustellen. Andererseits wird sie benötigt, um die zustandsabhängigen Grenzen der elektrisch realisierbaren Antriebsmomente zu bestimmen. Die Position \vec{h}_i der Radnabe i im Robotersystem ist konstruktiv festgelegt zu $\vec{h}_i = (\underline{h}_{ix} \quad \underline{h}_{iy} \quad \underline{h}_{iz})^T$. Übertragen in das globale Koordinatensystem folgt

$$\vec{h}_i = \vec{r} + B^{-1} \vec{h}_i. \quad (4.12)$$

Einfache Zeitableitung von (4.12) führt zur Nabengeschwindigkeit

$$\dot{\vec{h}}_i = \frac{d}{dt} \vec{h}_i = \dot{\vec{r}} + \dot{B}^{-1} \vec{h}_i = \begin{pmatrix} \dot{r}_x - (\underline{h}_{ix} \sin(\Psi) + \underline{h}_{iy} \cos(\Psi)) \dot{\Psi} \\ \dot{r}_y + (\underline{h}_{ix} \cos(\Psi) - \underline{h}_{iy} \sin(\Psi)) \dot{\Psi} \\ 0 \end{pmatrix} \dot{\Psi} \quad (4.13)$$

im globalen System.

Der Aufstandspunkt \vec{f}_i des Rades i im Nabensystem des Rades i mit dem Radradius r_i und dem Winkel des Aufstandspunktes $\theta_i = -\frac{\pi}{2}$ ist

$$\vec{f}_i = \begin{pmatrix} 0 \\ r_i \cos(\theta_i) \\ r_i \sin(\theta_i) \end{pmatrix}, \quad (4.14)$$

siehe Abbildung 4.5(d). Übertragen von \vec{f}_i ins globale System führt zu

$$\vec{f}_i = \vec{r} + B^{-1} \left(\vec{h}_i + B_i^{-1} \begin{pmatrix} 0 \\ r_i \cos(\theta_i) \\ r_i \sin(\theta_i) \end{pmatrix} \right). \quad (4.15)$$

Einfache Zeitableitung des Aufstandspunktes \vec{f}_i ergibt die Geschwindigkeit des Aufstandspunktes

$$\dot{\vec{f}}_i = \frac{d}{dt} \vec{f}_i = \begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ 0 \end{pmatrix} + \begin{pmatrix} \underline{h}_{ix} \sin(\Psi) + \underline{h}_{iy} \cos(\Psi) \\ \underline{h}_{ix} \cos(\Psi) - \underline{h}_{iy} \sin(\Psi) \\ 0 \end{pmatrix} \dot{\Psi} + r_i \begin{pmatrix} -\sin(\varphi_i + \Psi) \\ \cos(\varphi_i + \Psi) \\ 0 \end{pmatrix} \dot{\theta}_i \quad (4.16)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

im globalen System. Mit der Radrichtung $\vec{E}_{Hiy} = B^{-1} \underline{\underline{B}}_{Hi}^{-1} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T$ des Rades i (siehe Abbildung 4.5(d)) führt die Rollbedingung des Allseitenrades

$$0 = \dot{\underline{\underline{h}}}_i \vec{E}_{Hiy} \quad (4.17)$$

zur Winkelgeschwindigkeitskomponente $\omega_i := \dot{\theta}_i$ bezüglich der Achsrichtung:

$$\begin{aligned} \omega_i &= \frac{1}{r_i} \left(\sin(\varphi_i + \Psi) \dot{r}_x - \cos(\varphi_i + \Psi) \dot{r}_y - (\underline{\underline{h}}_{iy} \sin(\varphi_i) + \underline{\underline{h}}_{ix} \cos(\varphi_i)) \dot{\Psi} \right) \\ &= \frac{1}{r_i} \left(\sin(\varphi_i) \dot{\underline{\underline{r}}}_x - \cos(\varphi_i) \dot{\underline{\underline{r}}}_y - (\underline{\underline{h}}_{iy} \sin(\varphi_i) + \underline{\underline{h}}_{ix} \cos(\varphi_i)) \dot{\Psi} \right). \end{aligned} \quad (4.18)$$

Drehimpuls des Omnidirektionalrades

Das Omnidirektionalrad i wird als starrer Körper der Masse m_i und der Trägheitsmomente $\underline{\underline{J}}_{Hixx}$, $\underline{\underline{J}}_{Hiyy}$ und $\underline{\underline{J}}_{Hizz}$ um die Hauptträgheitsachsen \vec{E}_{Hix} , \vec{E}_{Hiy} und $\vec{E}_{Hiz} = \vec{e}_z$ betrachtet. Daraus ergibt sich der Drehimpuls \vec{L}_i des Rades i bezüglich der Radnabe im globalen System zu

$$\vec{L}_{Hi} = B^{-1} \underline{\underline{B}}_{Hi}^{-1} \begin{pmatrix} \underline{\underline{J}}_{Hixx} \omega_i \\ 0 \\ \underline{\underline{J}}_{Hizz} \dot{\Psi} \end{pmatrix} = \begin{pmatrix} L_{Hix} \\ L_{Hiy} \\ L_{Hiz} \end{pmatrix}, \quad (4.19)$$

wobei

$$\begin{aligned} L_{Hix} &= \frac{\underline{\underline{J}}_{Hixx}}{r_i} \left(\left((1 - 2 \sin^2(\Psi)) \cos(\varphi_i) \sin(\varphi_i) + (1 - 2 \sin^2(\varphi_i)) \cos(\Psi) \sin(\Psi) \right) \dot{r}_x \right. \\ &+ \left((1 - 2 \sin^2(\varphi_i)) \sin^2(\Psi) + 2 \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \sin(\Psi) + \sin^2(\varphi_i) - 1 \right) \dot{r}_y \\ &+ \left(\underline{\underline{h}}_{iy} (\sin^2(\varphi_i) \sin(\Psi) - \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi)) \right. \\ &\left. \left. + \underline{\underline{h}}_{ix} (\cos(\varphi_i) \sin(\varphi_i) \sin(\Psi) + (\sin^2(\varphi_i) - 1) \cos(\Psi)) \right) \dot{\Psi} \right), \end{aligned} \quad (4.20)$$

4. Regelung eines omnidirektionalen Antriebssystems

$$\begin{aligned}
 L_{Hiy} = \frac{\underline{\underline{J}}_{Hixx}}{r_i} & \left(\left(\left(1 - 2 \sin^2(\varphi_i) \right) \sin^2(\Psi) + 2 \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \sin(\Psi) + \sin^2(\varphi_i) \right) \dot{r}_x \right. \\
 & + \left(\left(2 \sin^2(\Psi) - 1 \right) \cos(\varphi_i) \sin(\varphi_i) + \left(2 \sin^2(\varphi_i) - 1 \right) \cos(\Psi) \sin(\Psi) \right) \dot{r}_y \\
 & - \left(\underline{h}_{iy} \left(\cos(\varphi_i) \sin(\varphi_i) \sin(\Psi) + \sin^2(\varphi_i) \cos(\Psi) \right) \right. \\
 & \left. \left. + \underline{h}_{ix} \left(\left(1 - \sin^2(\varphi_i) \right) \sin(\Psi) + \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \right) \right) \dot{\Psi} \right) \quad (4.21)
 \end{aligned}$$

und

$$L_{Hiz} = \underline{\underline{J}}_{Hizz} \dot{\Psi}. \quad (4.22)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

Für die zeitliche Ableitung des Drehimpulses $\dot{L}_{Hi} = (\dot{L}_{Hxi} \quad \dot{L}_{Hyi} \quad \dot{L}_{Hzi})^T$ des Rades i gilt

$$\begin{aligned} \dot{L}_{Hix} = & \frac{\tilde{J}^{Hixx}}{r_i} \left(\right. \\ & \left(-2 \cos(\varphi_i) \sin(\varphi_i) \sin^2(\Psi) + (1 - 2 \sin^2(\varphi_i)) \cos(\Psi) \sin(\Psi) + \cos(\varphi_i) \sin(\varphi_i) \right) \ddot{r}_x \\ & + \left((1 - 2 \sin^2(\varphi_i)) \sin^2(\Psi) + 2 \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \sin(\Psi) + \sin^2(\varphi_i) - 1 \right) \ddot{r}_y \\ & + \left(\underline{h}_{ix} \left(\cos(\varphi_i) \sin(\varphi_i) \sin(\Psi) + (\sin^2(\varphi_i) - 1) \cos(\Psi) \right) \right. \\ & \left. + \underline{h}_{iy} \left(\sin^2(\varphi_i) \sin(\Psi) - \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \right) \right) \ddot{\Psi} \\ & + \left(\left(\sin^2(\Psi) - \cos^2(\Psi) \right) \left(2 \sin^2(\varphi_i) - 1 \right) - 4 \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \sin(\Psi) \right) \dot{\Psi} \dot{r}_x \\ & - 2 \left(\left(\sin^2(\Psi) - \cos^2(\Psi) \right) \cos(\varphi_i) \sin(\varphi_i) + \left(2 \sin^2(\varphi_i) - 1 \right) \cos(\Psi) \sin(\Psi) \right) \dot{\Psi} \dot{r}_y \\ & + \left(\underline{h}_{ix} \left(\cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) + (1 - \sin^2(\varphi_i)) \sin(\Psi) \right) \right. \\ & \left. + \underline{h}_{iy} \left(\cos(\varphi_i) \sin(\varphi_i) \sin(\Psi) + \sin^2(\varphi_i) \cos(\Psi) \right) \right) \dot{\Psi}^2 \left. \right), \quad (4.23) \end{aligned}$$

4. Regelung eines omnidirektionalen Antriebssystems

$$\begin{aligned}
\dot{L}_{Hiy} = & -\frac{\underline{\underline{J}}_{Hixx}}{r_i} \left(\left(\left(2 \sin^2(\varphi_i) - 1 \right) \sin^2(\Psi) - 2 \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \sin(\Psi) - \sin^2(\varphi_i) \right) \ddot{r}_x \right. \\
& + \left(\left(1 - 2 \sin^2(\Psi) \right) \cos(\varphi_i) \sin(\varphi_i) + \left(1 - 2 \sin^2(\varphi_i) \right) \cos(\Psi) \sin(\Psi) \right) \ddot{r}_y \\
& + \left(\underline{h}_{iy} \left(\cos(\varphi_i) \sin(\varphi_i) \sin(\Psi) + \sin^2(\varphi_i) \cos(\Psi) \right) \right. \\
& \left. \left. + \underline{h}_{ix} \left((1 - \sin^2(\varphi_i)) \sin(\Psi) + \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \right) \right) \ddot{\Psi} \right. \\
& + 2 \left(\left(\sin^2(\Psi) - \cos^2(\Psi) \right) \cos(\varphi_i) \sin(\varphi_i) + \left(2 \sin^2(\varphi_i) - 1 \right) \cos(\Psi) \sin(\Psi) \right) \dot{\Psi} \dot{r}_x \\
& + \left(\left(\cos^2(\Psi) - \sin^2(\Psi) \right) \left(1 - 2 \sin^2(\varphi_i) \right) - 4 \cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) \sin(\Psi) \right) \dot{\Psi} \dot{r}_y \\
& + \left(\underline{h}_{iy} \left(\cos(\varphi_i) \sin(\varphi_i) \cos(\Psi) - \sin^2(\varphi_i) \sin(\Psi) \right) \right. \\
& \left. \left. + \underline{h}_{ix} \left((1 - \sin^2(\varphi_i)) \cos(\Psi) - \cos(\varphi_i) \sin(\varphi_i) \sin(\Psi) \right) \right) \dot{\Psi}^2 \right) \quad (4.24)
\end{aligned}$$

und

$$\dot{L}_{Hiz} = \underline{\underline{J}}_{Hizz} \ddot{\Psi}. \quad (4.25)$$

Mit $\dot{\underline{\underline{L}}}_{Hi} = \begin{pmatrix} \dot{\underline{L}}_{Hix} & \dot{\underline{L}}_{Hiy} & \dot{\underline{L}}_{Hiz} \end{pmatrix}^T = B \dot{\underline{\underline{L}}}_{Hi}$ wird die Zeitableitung des Rad-drehimpulses in das Robotersystem projiziert. Dabei wird die Geschwindigkeit

4.1. Modellierung eines omnidirektional getriebenen Roboters

des Roboters im globalen System durch deren, im Robotersystem messbaren, Projektion zu $B^{-1}\dot{\vec{r}} = \dot{\vec{r}}$ ersetzt. Es gilt

$$\begin{aligned} \dot{\underline{\underline{L}}}_{Hix} = \frac{\underline{\underline{J}}_{Hixx}}{r_i} & \left(\cos(\varphi_i) \sin(\varphi_i) \dot{\underline{\underline{r}}}_x - \cos^2(\varphi_i) \dot{\underline{\underline{r}}}_y \right. \\ & - \cos(\varphi_i) \left(\sin(\varphi_i) \underline{\underline{h}}_{iy} + \cos(\varphi_i) \underline{\underline{h}}_{ix} \right) \ddot{\Psi} \\ & + \left(2 \cos^2(\varphi_i) - 1 \right) \dot{\underline{\underline{r}}}_x \dot{\Psi} + 2 \cos(\varphi_i) \sin(\varphi_i) \dot{\underline{\underline{r}}}_y \dot{\Psi} \\ & \left. + \left((1 - \cos^2(\varphi_i)) \underline{\underline{h}}_{iy} + \cos(\varphi_i) \sin(\varphi_i) \underline{\underline{h}}_{ix} \right) \dot{\Psi}^2 \right), \quad (4.26) \end{aligned}$$

$$\begin{aligned} \dot{\underline{\underline{L}}}_{Hiy} = \frac{\underline{\underline{J}}_{Hixx}}{r_i} & \left(\left(1 - \cos^2(\varphi_i) \right) \dot{\underline{\underline{r}}}_x - \cos(\varphi_i) \sin(\varphi_i) \dot{\underline{\underline{r}}}_y \right. \\ & + \left((\cos^2(\varphi_i) - 1) \underline{\underline{h}}_{iy} - \cos(\varphi_i) \sin(\varphi_i) \underline{\underline{h}}_{ix} \right) \ddot{\Psi} \\ & + 2 \cos(\varphi_i) \sin(\varphi_i) \dot{\underline{\underline{r}}}_x \dot{\Psi} + \left(1 - 2 \cos^2(\varphi_i) \right) \dot{\underline{\underline{r}}}_y \dot{\Psi} \\ & \left. - \left(\cos(\varphi_i) \sin(\varphi_i) \underline{\underline{h}}_{iy} + \cos^2(\varphi_i) \underline{\underline{h}}_{ix} \right) \dot{\Psi}^2 \right) \quad (4.27) \end{aligned}$$

und

$$\dot{\underline{\underline{L}}}_{Hiz} = \underline{\underline{J}}_{Hizz} \ddot{\Psi}. \quad (4.28)$$

Impuls des Omnidirektionalrades

Mit (4.13) folgt der Impuls \vec{p}_{Hi} des Rades i der Masse m_{Hi} zu

$$\vec{p}_{Hi} = m_{Hi} \dot{\vec{h}}_i = m_{Hi} \begin{pmatrix} \dot{r}_x - (\underline{\underline{h}}_{ix} \sin(\Psi) + \underline{\underline{h}}_{iy} \cos(\Psi)) \dot{\Psi} \\ \dot{r}_y - (\underline{\underline{h}}_{iy} \sin(\Psi) - \underline{\underline{h}}_{ix} \cos(\Psi)) \dot{\Psi} \\ 0 \end{pmatrix}. \quad (4.29)$$

4. Regelung eines omnidirektionalen Antriebssystems

Für konstante Radmasse m_{Hi} ergibt sich die zeitliche Ableitung $\dot{\vec{p}}_{Hi}$ des Radimpulses zu

$$\dot{\vec{p}}_{Hi} = m_{Hi} \begin{pmatrix} \ddot{r}_x - (\underline{h}_{ix} \sin(\Psi) + \underline{h}_{iy} \cos(\Psi))\ddot{\Psi} + (\underline{h}_{iy} \sin(\Psi) - \underline{h}_{ix} \cos(\Psi))\dot{\Psi}^2 \\ \ddot{r}_y + (\underline{h}_{ix} \cos(\Psi) - \underline{h}_{iy} \sin(\Psi))\ddot{\Psi} - (\underline{h}_{ix} \sin(\Psi) + \underline{h}_{iy} \cos(\Psi))\dot{\Psi}^2 \\ 0 \end{pmatrix}. \quad (4.30)$$

Mit $\dot{\vec{p}}_{Hi} = B\dot{\vec{p}}_{Hi}$ wird die Zeitableitung des Radimpulses in das Robotersystem projiziert. Die Beschleunigung des Roboters im globalen System wird durch deren, im Robotersystem messbaren, Projektion zu $B^{-1}\ddot{\underline{r}} = \ddot{\underline{r}}$ ersetzt. Es gilt

$$\ddot{\underline{p}}_{Hi} = m_{Hi} \begin{pmatrix} \ddot{r}_x - \underline{h}_{iy}\ddot{\Psi} - \underline{h}_{ix}\dot{\Psi}^2 \\ \ddot{r}_y + \underline{h}_{ix}\ddot{\Psi} - \underline{h}_{iy}\dot{\Psi}^2 \\ 0 \end{pmatrix}. \quad (4.31)$$

Kräfte und Momente am Omnidirektionalrad

Das Omnidirektionalrad nach [Gra19] ist eine Sonderform des Rades, bei der senkrecht zum Kreisumfang kleine Rollen, die Kranzräder, montiert sind, siehe Abbildung 4.1 auf Seite 73. Bei Bewegung des Rades entlang seiner Achse \vec{E}_{Hix} können diese ohne kinematischen Zwang abrollen. Bei Bewegung des Rades entlang der Radrichtung \vec{E}_{Hiy} ist dieses Abrollen hingegen nicht möglich – über die Rollbedingung (4.17) erfolgt die Kopplung zwischen Bewegung und Winkelgeschwindigkeit durch (4.18).

Am Rad wirken im Nabensystem die Zwangskraft $\vec{\tilde{Z}}_{Hi} = (0 \ 0 \ Z_i)^T$ des Kontaktes zum Boden, die Gewichtskraft $\vec{\tilde{G}}_{Hi} = (0 \ 0 \ -m_{Hi}g)^T$, die Haftreibungskraft in Radrichtung $\vec{\tilde{D}}_{Hi} = (0 \ D_i \ 0)^T$ und die Rollreibungskraft des Omnidirektionalrades in Achsrichtung $\vec{\tilde{H}}_{Hi} = (H_i \ 0 \ 0)^T$. Siehe Abbildung 4.6.

Für die Haftreibungskraft $\vec{\tilde{D}}_{Hi}$ gilt die Haftreibungsgrenze

$$\mu_i Z_i \geq |D_i| \quad (4.32)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

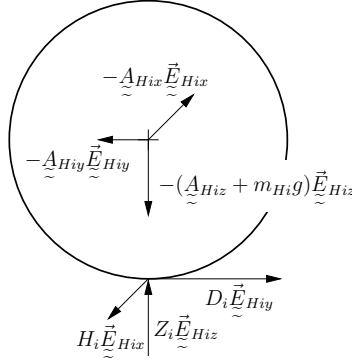


Abbildung 4.6.: Kräfte am Omnidirektionalrad

mit dem Haftreibungskoeffizienten μ_i . Wird die Rollreibungskraft \vec{H}_{Hi} als coulombsche Reibung mit der in Achsrichtung \vec{E}_{Hix} projizierten Nabengeschwindigkeit $\dot{\vec{h}}_i$ zu

$$H_i = -\mu_{gi} Z_i \text{sign} \left(\dot{\vec{h}}_i \vec{E}_{Hix} \right) \quad (4.33)$$

modelliert, so ergibt sich eine Unstetigkeit beim Nulldurchgang der Signumfunktion. Um hieraus erwachsende Probleme im praktischen Einsatz des Modells als Grundlage eines Reglers zu vermeiden, wird nach [DAH00] eine Approximation der Reibkraft zu

$$H_i = -\mu_{gi} Z_i \tanh \left(\frac{1}{v_{0i}} \dot{\vec{h}}_i \vec{E}_{Hix} \right) = k_i Z_i \quad (4.34)$$

vorgenommen. Dabei ist μ_{gi} der Rollreibungskoeffizient und v_{0i} ein Skalierungsfaktor². Der Term $-\mu_{gi} \tanh \left(\frac{1}{v_{0i}} \dot{\vec{h}}_i \vec{E}_{Hix} \right)$ wird zu einem geschwindigkeitsabhängigen Faktor k_i für die weitere Verwendung zusammengefasst, wobei

$$\dot{\vec{h}}_i \vec{E}_{Hix} = \cos \varphi_i \dot{\vec{x}} + \sin \varphi_i \dot{\vec{y}} + \left(\sin \varphi_i \vec{h}_{ix} - \cos \varphi_i \vec{h}_{iy} \right) \dot{\Psi}. \quad (4.35)$$

² Der Skalierungsfaktor v_{0i} wird nach [Mar07] zu $v_{0i} \approx 10^{-5} \frac{m}{s}$ angenommen.

4. Regelung eines omnidirektionalen Antriebssystems

Die Summe $\vec{\underline{F}}_{Hi}$ aller am Rad angreifenden Kräfte bezüglich des Robotersystems ist

$$\begin{aligned}\vec{\underline{F}}_{Hi} &= \underline{B}_{Hi}^{-1} \left(\vec{\underline{Z}}_{Hi} + \vec{\underline{G}}_{Hi} + \vec{\underline{D}}_{Hi} + \vec{\underline{H}}_{Hi} \right) - \vec{\underline{A}}_{Hi} \\ &= \begin{pmatrix} 0 \\ 0 \\ -gm_{Hi} \end{pmatrix} + \begin{pmatrix} k_i \cos(\varphi_i) \\ k_i \sin(\varphi_i) \\ 1 \end{pmatrix} Z_i + \begin{pmatrix} -\sin(\varphi_i) \\ \cos(\varphi_i) \\ 0 \end{pmatrix} D_i - \vec{\underline{A}}_{Hi}. \quad (4.36)\end{aligned}$$

Die in das Robotersystem projizierte Summe $\vec{\underline{M}}_{Hi}$ aller am Rad bezüglich der Nabe angreifenden Momente ist

$$\begin{aligned}\vec{\underline{M}}_{Hi} &= \underline{B}_{Hi}^{-1} \left(\begin{pmatrix} 0 \\ 0 \\ -r_i \end{pmatrix} \times \left(\vec{\underline{D}}_{Hi} + \vec{\underline{H}}_{Hi} \right) \right) - \vec{\underline{T}}_{Hi} \\ &= \begin{pmatrix} k_i r_i \sin(\varphi_i) \\ -k_i r_i \cos(\varphi_i) \\ 0 \end{pmatrix} Z_i + \begin{pmatrix} r_i \cos(\varphi_i) \\ r_i \sin(\varphi_i) \\ 0 \end{pmatrix} D_i - \vec{\underline{T}}_{Hi}. \quad (4.37)\end{aligned}$$

Impulssatz für ein Rad

Der Impulssatz für das Rad i lautet

$$\dot{\vec{p}}_{Hi} = \vec{F}_{Hi}. \quad (4.38)$$

Beidseitiges multiplizieren mit B führt zu

$$B\dot{\vec{p}}_{Hi} = \dot{\vec{p}}_{Hi} = B\vec{F}_{Hi} = \vec{\underline{F}}_{Hi}, \quad (4.39)$$

wobei $\dot{\vec{p}}_{Hi}$ durch (4.31) und $\vec{\underline{F}}_{Hi}$ durch (4.36) bestimmt werden. Somit ergibt sich die Schnittkraft $\vec{\underline{A}}_{Hi}$ zu

$$\vec{\underline{A}}_{Hi} = m_{Hi} \begin{pmatrix} -\ddot{x} + \underline{h}_{iy} \ddot{\Psi} + \underline{h}_{ix} \dot{\Psi}^2 \\ -\ddot{y} - \underline{h}_{ix} \ddot{\Psi} + \underline{h}_{iy} \dot{\Psi}^2 \\ -g \end{pmatrix} + \begin{pmatrix} k_i \cos(\varphi_i) \\ k_i \sin(\varphi_i) \\ 1 \end{pmatrix} Z_i + \begin{pmatrix} -\sin(\varphi_i) \\ \cos(\varphi_i) \\ 0 \end{pmatrix} D_i. \quad (4.40)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

Drehimpulssatz für ein Rad

Der Drehimpulssatz für das Rad i lautet

$$\dot{\vec{L}}_{Hi} = \vec{M}_{Hi}. \quad (4.41)$$

Beidseitiges multiplizieren mit B führt zu

$$B\dot{\vec{L}}_{Hi} = \dot{\vec{L}}_{Hi} = B\vec{M}_{Hi} = \vec{M}_{Hi}. \quad (4.42)$$

Die in das Robotersystem projizierte zeitliche Drehimpulsänderung $\dot{\vec{L}}_{Hi}$ steht dabei aus (4.26)-(4.28) und die Momentensumme \vec{M}_{Hi} bezüglich des Robotersystems aus (4.37) zur Verfügung.

Das Schnittmoment $\vec{T}_{Hi} = (\underline{T}_{Hix} \quad \underline{T}_{Hiy} \quad \underline{T}_{Hiz})^T$ ergibt sich damit zu

$$\begin{aligned} \underline{T}_{Hix} = & \frac{\underline{J}_{Hixx}}{2r_i} \left(-\sin(2\varphi_i) \ddot{\underline{x}} + (\cos(2\varphi_i) + 1) \ddot{\underline{y}} \right. \\ & + \left(\sin(2\varphi_i) \underline{h}_{iy} + (\cos(2\varphi_i) + 1) \underline{h}_{ix} \right) \ddot{\Psi} + \left((\cos(2\varphi_i) - 1) \underline{h}_{iy} - \sin(2\varphi_i) \underline{h}_{ix} \right) \dot{\Psi}^2 \\ & \left. - 2 \cos(2\varphi_i) \dot{\underline{x}} \dot{\Psi} - 2 \sin(2\varphi_i) \dot{\underline{y}} \dot{\Psi} \right) + r_i k_i \sin(\varphi_i) Z_i + r_i \cos(\varphi_i) D_i, \quad (4.43) \end{aligned}$$

$$\begin{aligned} \underline{T}_{Hiy} = & \frac{\underline{J}_{Hixx}}{2r_i} \left((\cos(2\varphi_i) - 1) \ddot{\underline{x}} + \sin(2\varphi_i) \ddot{\underline{y}} \right. \\ & + \left((1 - \cos(2\varphi_i)) \underline{h}_{iy} + \sin(2\varphi_i) \underline{h}_{ix} \right) \ddot{\Psi} + \left(\sin(2\varphi_i) \underline{h}_{iy} + (\cos(2\varphi_i) + 1) \underline{h}_{ix} \right) \dot{\Psi}^2 \\ & \left. - 2 \sin(2\varphi_i) \dot{\underline{x}} \dot{\Psi} + 2 \cos(2\varphi_i) \dot{\underline{y}} \dot{\Psi} \right) - r_i k_i \cos(\varphi_i) Z_i + r_i \sin(\varphi_i) D_i \quad (4.44) \end{aligned}$$

und

$$\underline{T}_{Hiz} = -\underline{J}_{Hiz} \ddot{\Psi}. \quad (4.45)$$

4. Regelung eines omnidirektionalen Antriebssystems

4.1.5. Dynamik des Antriebes

Geschwindigkeit und Winkelgeschwindigkeit des Antriebes

Der Schwerpunkt $\underline{\tilde{s}}_i$ der rotierenden Antriebsmasse ist im Robotersystem gegeben zu

$\underline{\tilde{s}}_i = (\underline{\tilde{s}}_{xi} \quad \underline{\tilde{s}}_{yi} \quad \underline{\tilde{s}}_{zi})^T$. Daraus folgt der Antriebsschwerpunkt \underline{s}_i^* zu

$$\underline{s}_i^* = \underline{r} + B^{-1} \underline{\tilde{s}}_i \quad (4.46)$$

im globalen System und analog zu (4.13) dessen Geschwindigkeit

$$\dot{\underline{s}}_i^* = \frac{d}{dt} \underline{s}_i^* = \dot{\underline{r}} + \dot{B}^{-1} \underline{\tilde{s}}_i = \begin{pmatrix} \dot{r}_x - (\underline{\tilde{s}}_{xi} \sin(\Psi) + \underline{\tilde{s}}_{yi} \cos(\Psi)) \dot{\Psi} \\ \dot{r}_y + (\underline{\tilde{s}}_{xi} \cos(\Psi) - \underline{\tilde{s}}_{yi} \sin(\Psi)) \dot{\Psi} \\ 0 \end{pmatrix}. \quad (4.47)$$

Mit Hilfe der Übersetzung γ_i leite sich die Winkelgeschwindigkeit ξ_i des Antriebes um seine x -Achse direkt aus der korrespondierenden Radwinkelgeschwindigkeitskomponente ω_i ab:

$$\xi_i = \gamma_i \omega_i. \quad (4.48)$$

Drehimpuls des Antriebes

Analog zum Omnidirektionalrad 4.1.4 wird der rotierende Teil des Antriebes i als starrer Körper der Masse m_{Si} und der Trägheitsmomente $\underline{\tilde{J}}_{Sidx}$, $\underline{\tilde{J}}_{Sidy}$ und $\underline{\tilde{J}}_{Siz}$ um die Hauptträgheitsachsen $\underline{\tilde{E}}_{Six}$, $\underline{\tilde{E}}_{Siy}$ und $\underline{\tilde{E}}_{Siz} = \underline{\tilde{e}}_z$ betrachtet. Der Drehimpuls $\underline{\tilde{L}}_{Si}$ des Antriebes i bezüglich seines Schwerpunktes ist

$$\underline{\tilde{L}}_{Si} = B^{-1} \underline{\tilde{B}}_{Si}^{-1} \begin{pmatrix} \underline{\tilde{J}}_{Sidx} \xi_i \\ 0 \\ \underline{\tilde{J}}_{Siz} \dot{\Psi} \end{pmatrix}. \quad (4.49)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

Die zeitliche Ableitung $\dot{\underline{L}}_{Si}$ des Antriebsdrehimpulses wird analog zur zeitlichen Ableitung des Raddrehimpulses (4.26)-(4.28) zu $\dot{\underline{L}}_{Si} = (\dot{\underline{L}}_{S_{ix}} \quad \dot{\underline{L}}_{S_{iy}} \quad \dot{\underline{L}}_{S_{iz}})^T$ in das Robotersystem projiziert mit

$$\begin{aligned} \dot{\underline{L}}_{S_{ix}} = \frac{\gamma_i \underline{\tilde{J}}_{S_{ixx}}}{r_i} & \left(\cos(\beta_i) \sin(\varphi_i) \ddot{r}_x - \cos(\beta_i) \cos(\varphi_i) \ddot{r}_y \right. \\ & - \cos(\beta_i) \left(\sin(\varphi_i) \underline{h}_{iy} + \cos(\varphi_i) \underline{h}_{ix} \right) \ddot{\Psi} \\ & + \left(\cos(\beta_i) \cos(\varphi_i) - \sin(\beta_i) \sin(\varphi_i) \right) \dot{r}_x \dot{\Psi} \\ & + \left(\cos(\beta_i) \sin(\varphi_i) + \sin(\beta_i) \cos(\varphi_i) \right) \dot{r}_y \dot{\Psi} \\ & \left. + \left(\underline{h}_{ix} \sin(\beta_i) \cos(\varphi_i) + \underline{h}_{iy} \sin(\beta_i) \sin(\varphi_i) \right) \dot{\Psi}^2 \right), \quad (4.50) \end{aligned}$$

$$\begin{aligned} \dot{\underline{L}}_{S_{iy}} = \frac{\gamma_i \underline{\tilde{J}}_{S_{iyy}}}{r_i} & \left(\sin(\beta_i) \sin(\varphi_i) \ddot{r}_x - \sin(\beta_i) \cos(\varphi_i) \ddot{r}_y \right. \\ & - \sin(\beta_i) \left(\sin(\varphi_i) \underline{h}_{iy} + \cos(\varphi_i) \underline{h}_{ix} \right) \ddot{\Psi} \\ & + \left(\sin(\beta_i) \cos(\varphi_i) + \cos(\beta_i) \sin(\varphi_i) \right) \dot{r}_x \dot{\Psi} \\ & + \left(\sin(\beta_i) \sin(\varphi_i) - \cos(\beta_i) \cos(\varphi_i) \right) \dot{r}_y \dot{\Psi} \\ & \left. - \left(\underline{h}_{ix} \cos(\beta_i) \cos(\varphi_i) + \underline{h}_{iy} \cos(\beta_i) \sin(\varphi_i) \right) \dot{\Psi}^2 \right) \quad (4.51) \end{aligned}$$

und

$$\dot{\underline{L}}_{S_{iz}} = \underline{\tilde{J}}_{S_{izz}} \ddot{\Psi}. \quad (4.52)$$

Impuls des Antriebes

Mit (4.47) folgt der Impuls \vec{p}_{Si} der rotierenden Antriebsmasse m_{Si} zu

$$\vec{p}_{Si} = m_{Si} \begin{pmatrix} \dot{r}_x - \left(\underline{s}_{ix} \sin(\Psi) + \underline{s}_{iy} \cos(\Psi) \right) \dot{\Psi} \\ \dot{r}_y - \left(\underline{s}_{iy} \sin(\Psi) - \underline{s}_{ix} \cos(\Psi) \right) \dot{\Psi} \\ 0 \end{pmatrix}. \quad (4.53)$$

4. Regelung eines omnidirektionalen Antriebssystems

Die zeitliche Ableitung $\dot{\vec{p}}_{Si}$ des Impulses des Antriebes wird analog zur zeitlichen Ableitung des Impulses des Rades (4.31) ins Robotersystem projiziert zu

$$\dot{\vec{p}}_{Si} = m_{Si} \begin{pmatrix} \ddot{\tilde{x}} - \tilde{s}_{iy}\ddot{\Psi} - \tilde{s}_{ix}\dot{\Psi}^2 \\ \ddot{\tilde{y}} + \tilde{s}_{ix}\ddot{\Psi} - \tilde{s}_{iy}\dot{\Psi}^2 \\ 0 \end{pmatrix}. \quad (4.54)$$

Kräfte und Momente am Antrieb

Am Antrieb wirkt im Robotersystem die Gewichtskraft $\vec{G}_{Si} = (0 \ 0 \ -m_{Si}g)^T$ und die negative Schnittkraft \vec{A}_{Si} . Somit ist die Summe \vec{F}_{Si} aller am Antrieb angreifenden Kräfte bezüglich des Robotersystems

$$\vec{F}_{Si} = - \begin{pmatrix} A_{S ix} & A_{S iy} & A_{S iz} + m_{Si}g \end{pmatrix}^T. \quad (4.55)$$

Das einzige am Rotor wirksame Moment ist das negative Schnittmoment, die Summe \vec{M}_{Si} aller am Antrieb angreifenden Momente bezüglich des Robotersystems ist also

$$\vec{M}_{Si} = -\vec{T}_{Si}. \quad (4.56)$$

Für die später folgende Bestimmung des Antriebsmomentes wird für jedes Rad und seinen Antrieb die Summe der auf die Rotationsachsen \vec{E}_{Hix} bzw. $\vec{E}_{S ix}$ projizierten Momentensummen (4.56) bzw. (4.37) gebildet.

Impulssatz für einen Antrieb

Der Impulssatz für den Antrieb i lautet

$$\dot{\vec{p}}_{Si} = \vec{F}_{Si}. \quad (4.57)$$

Beidseitiges multiplizieren mit B führt zu

$$B\dot{\vec{p}}_{Si} = \dot{\vec{p}}_{Si} = B\vec{F}_{Si} = \vec{F}_{Si}, \quad (4.58)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

wobei die in das Robotersystem projizierte Impulsänderung $\dot{\tilde{\underline{p}}}_{Si}$ mit (4.54) und die Kraftsumme $\tilde{\underline{F}}_{Si}$ des Antriebes mit (4.55) festgelegt ist. Die Schnittkraft $\tilde{\underline{A}}_{Si}$ ergibt sich damit zu

$$\tilde{\underline{A}}_{Si} = m_{Si} \begin{pmatrix} -\ddot{\tilde{r}}_x + \underline{s}_{iy}\ddot{\Psi} + \underline{s}_{ix}\dot{\Psi}^2 \\ -\ddot{\tilde{r}}_y - \underline{s}_{ix}\ddot{\Psi} + \underline{s}_{iy}\dot{\Psi}^2 \\ -g \end{pmatrix}. \quad (4.59)$$

Drehimpulssatz für einen Antrieb

Der Drehimpulssatz für den Antrieb i lautet

$$\dot{\tilde{\underline{L}}}_{Si} = \tilde{\underline{M}}_{Si}. \quad (4.60)$$

Beidseitiges multiplizieren mit B führt zu

$$B\dot{\tilde{\underline{L}}}_{Si} = \dot{\tilde{\underline{L}}}_{Si} = B\tilde{\underline{M}}_{Si} = \tilde{\underline{M}}_{Si}, \quad (4.61)$$

wobei die Projektion $\dot{\tilde{\underline{L}}}_{Si}$ der zeitlichen Ableitung des Antriebsdrehimpulses ins Robotersystem durch (4.50)-(4.52) und die Momentensumme $\tilde{\underline{M}}_{Si}$ durch (4.56) festgelegt ist.

Hieraus ergibt sich das Schnittmoment $\tilde{\underline{T}}_{Si} = (\underline{T}_{S ix} \quad \underline{T}_{S iy} \quad \underline{T}_{S iz})^T$ mit

$$\begin{aligned} \underline{T}_{S ix} = \frac{\gamma_i \underline{J}_{S i x x}}{2r_i} & \left(- \left(\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i) \right) \underline{\ddot{r}}_x \right. \\ & + \left(\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i) \right) \underline{\ddot{r}}_y \\ & + \left(\underline{h}_{yi} \left(\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i) \right) \right. \\ & \left. + \underline{h}_{xi} \left(\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i) - \sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i) \right) \right) \ddot{\Psi} \\ & \left. - 2 \cos(\varphi_i + \beta_i) \underline{\dot{r}}_x \dot{\Psi} - 2 \sin(\varphi_i + \beta_i) \underline{\dot{r}}_y \dot{\Psi} + \underline{h}_{yi} \left(\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i) \right) \dot{\Psi}^2 \right), \end{aligned} \quad (4.62)$$

4. Regelung eines omnidirektionalen Antriebssystems

$$\begin{aligned}
 \underline{T}_{Siy} = \frac{\gamma_i \underline{J}_{S_{ixx}}}{2r_i} & \left(\left(\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i) \right) \underline{\ddot{r}}_x \right. \\
 & + \left(\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i) \right) \underline{\ddot{r}}_y \\
 & + \left(\underline{h}_{yi} \left(\cos(\varphi_i - \beta_i) - \cos(\varphi_i + \beta_i) \right) \right. \\
 & + \underline{h}_{xi} \left(\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i) \right) \left. \right) \ddot{\Psi} \\
 & - 2 \sin(\varphi_i + \beta_i) \underline{\dot{r}}_x \dot{\Psi} + 2 \cos(\varphi_i + \beta_i) \underline{\dot{r}}_y \dot{\Psi} \\
 & + \left(\underline{h}_{xi} \left(\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i) \right) + \underline{h}_{yi} \left(\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i) \right) \right) \dot{\Psi}^2 \Big)
 \end{aligned} \tag{4.63}$$

und

$$\underline{T}_{Siz} = -\underline{J}_{S_{izz}} \ddot{\Psi}. \tag{4.64}$$

4.1.6. Dynamik des Grundkörpers

Drehimpuls des Grundkörpers

Wie Rad und rotierender Teil des Antriebes wird auch der Grundkörper der Masse m_B als starrer Körper betrachtet. Dabei sind \underline{J}_{Bxx} , \underline{J}_{Byy} und \underline{J}_{Bzz} die Trägheitsmomente des Grundkörpers um die Hauptträgheitsachsen $\underline{\vec{e}}_{Bx}$, $\underline{\vec{e}}_{By}$ und $\underline{\vec{e}}_{Bz}$. Der Drehimpuls³ $\underline{\vec{L}}_B$ des Grundkörpers sei

$$\underline{\vec{L}}_B = B^{-1} B_B^{-1} \begin{pmatrix} 0 & 0 & \underline{J}_{Bzz} \dot{\Psi} \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & \underline{J}_{Bzz} \dot{\Psi} \end{pmatrix}^T, \tag{4.65}$$

die zeitliche Ableitung $\dot{\underline{\vec{L}}}_B$ des Drehimpulses ist

$$\dot{\underline{\vec{L}}}_B = \begin{pmatrix} 0 & 0 & \underline{J}_{Bzz} \ddot{\Psi} \end{pmatrix}^T. \tag{4.66}$$

Die Projektion der zeitlichen Ableitung des Drehimpulses ins Robotersystem führt zu

$$\dot{\underline{\vec{L}}}_B = \begin{pmatrix} 0 & 0 & \underline{J}_{Bzz} \ddot{\Psi} \end{pmatrix}^T. \tag{4.67}$$

³Es wird gefordert, dass alle Räder immer Bodenkontakt haben, Roll- und Nickbewegungen um $\underline{\vec{e}}_{By}$ bzw. $\underline{\vec{e}}_{Bx}$ werden gesperrt.

4.1. Modellierung eines omnidirektional getriebenen Roboters

Impuls des Grundkörpers

Die Position \vec{b} des Grundkörpers im Robotersystem ist konstruktiv festgelegt zu $\vec{b} = (\underline{b}_x \quad \underline{b}_y \quad \underline{b}_z)$. Übertragen in das globale System folgt

$$\vec{b} = \vec{r} + B^{-1}\vec{b}. \quad (4.68)$$

Die Geschwindigkeit des Grundkörpers ist somit

$$\dot{\vec{b}} = \dot{\vec{r}} + \dot{B}^{-1}\vec{b}. \quad (4.69)$$

Der Impuls \vec{p}_B des Grundkörpers der Masse m_B ist

$$\vec{p}_B = m_B \begin{pmatrix} \dot{r}_x - \left(\sin(\Psi)\underline{b}_x + \cos(\Psi)\underline{b}_y \right) \dot{\Psi} \\ \dot{r}_y + \left(\cos(\Psi)\underline{b}_x - \sin(\Psi)\underline{b}_y \right) \dot{\Psi} \\ 0 \end{pmatrix}. \quad (4.70)$$

Für konstante Masse ist die zeitliche Ableitung $\dot{\vec{p}}_B$ des Impulses

$$\dot{\vec{p}}_B = m_B \begin{pmatrix} \ddot{r}_x - \left(\cos(\Psi)\underline{b}_y + \sin(\Psi)\underline{b}_x \right) \ddot{\Psi} + \left(\sin(\Psi)\underline{b}_y - \cos(\Psi)\underline{b}_x \right) \dot{\Psi}^2 \\ \ddot{r}_y + \left(\cos(\Psi)\underline{b}_x - \sin(\Psi)\underline{b}_y \right) \ddot{\Psi} - \left(\cos(\Psi)\underline{b}_y + \sin(\Psi)\underline{b}_x \right) \dot{\Psi}^2 \\ 0 \end{pmatrix}. \quad (4.71)$$

Projektion der Impulsänderung ins Robotersystem unter Verwendung der ins Robotersystem projizierten Beschleunigung führt zu

$$\underline{\dot{\vec{p}}}_B = m_B \begin{pmatrix} \ddot{\underline{r}}_x - \underline{b}_y \ddot{\Psi} - \underline{b}_x \dot{\Psi}^2 \\ \ddot{\underline{r}}_y + \underline{b}_x \ddot{\Psi} - \underline{b}_y \dot{\Psi}^2 \\ 0 \end{pmatrix}. \quad (4.72)$$

Kräfte und Momente am Grundkörper

Am Grundkörper greifen die positiven Schnittkräfte \vec{A}_{Hi} und \vec{A}_{Si} sowie die Gewichtskraft

$\vec{G}_B = (0 \quad 0 \quad -gm_B)^T$ an, die Kraftsumme \vec{F}_B des Grundkörpers ist

$$\vec{F}_B = \vec{G}_B + \sum_i \vec{A}_{Hi} + \sum_i \vec{A}_{Si}. \quad (4.73)$$

4. Regelung eines omnidirektionalen Antriebssystems

Die am Grundkörper wirkenden Momente sind einerseits die positiven Schnittmomente \vec{T}_{Hi} und \vec{T}_{Si} , andererseits die an den Vektoren $\vec{h}_i - \vec{b}$ und $\vec{s}_i - \vec{b}$ wirkenden positiven Schnittkräfte \vec{A}_{Hi} und \vec{A}_{Si} . Die Summe \vec{M}_B aller am Grundkörper angreifenden Momente ist somit

$$\vec{M}_B = \sum_i [(\vec{h}_i - \vec{b}) \times \vec{A}_{Hi}] + \sum_i [(\vec{s}_i - \vec{b}) \times \vec{A}_{Si}] + \sum_i \vec{T}_{Hi} + \sum_i \vec{T}_{Si}. \quad (4.74)$$

Impulssatz für den Grundkörper

Der Impulssatz für den Grundkörper lautet

$$\dot{\vec{p}}_B = \vec{F}_B. \quad (4.75)$$

Beidseitiges multiplizieren mit B führt zu

$$B\dot{\vec{p}}_B = \dot{\vec{p}}_B = B\vec{F}_B = \vec{F}_B, \quad (4.76)$$

hierbei ist die Projektion $\dot{\vec{p}}_B$ der zeitlichen Impulsableitung durch (4.72) und die Kraftsumme \vec{F}_B durch (4.73) festgelegt. Weiterhin stehen die für die Bestimmung der Kraftsumme notwendigen Schnittkräfte mit (4.40) und (4.59) zur Verfügung. Damit ergeben sich aus dem Impulssatz für den Grundkörper die Gleichungen

$$\begin{aligned} & m_B \ddot{\underline{r}}_x - m_B \underline{b}_y \ddot{\Psi} - m_B \underline{b}_x \dot{\Psi}^2 \\ &= - \sum_i \left[m_{Hi} + m_{Si} \right] \ddot{\underline{r}}_x + \sum_i \left[m_{Hi} \underline{h}_{iy} + m_{Si} \underline{s}_{iy} \right] \ddot{\Psi} + \sum_i \left[m_{Hi} \underline{h}_{ix} + m_{Si} \underline{s}_{ix} \right] \dot{\Psi}^2 \\ &\quad + \sum_i \left[k_i \cos(\varphi_i) Z_i \right] - \sum_i \left[\sin(\varphi_i) D_i \right], \end{aligned} \quad (4.77)$$

$$\begin{aligned} & m_B \ddot{\underline{r}}_y + m_B \underline{b}_x \ddot{\Psi} - m_B \underline{b}_y \dot{\Psi}^2 \\ &= - \sum_i \left[m_{Hi} + m_{Si} \right] \ddot{\underline{r}}_y - \sum_i \left[m_{Hi} \underline{h}_{ix} + m_{Si} \underline{s}_{ix} \right] \ddot{\Psi} + \sum_i \left[m_{Hi} \underline{h}_{iy} + m_{Si} \underline{s}_{iy} \right] \dot{\Psi}^2 \\ &\quad + \sum_i \left[k_i \sin(\varphi_i) Z_i \right] + \sum_i \left[\cos(\varphi_i) D_i \right] \end{aligned} \quad (4.78)$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

und

$$0 = \sum_i Z_i - \left(\sum_i [m_{Hi} + m_{Si}] + m_B \right) g. \quad (4.79)$$

Drehimpulssatz für den Grundkörper

Der Drehimpulssatz für den Grundkörper lautet

$$\dot{\vec{L}}_B = \vec{M}_B. \quad (4.80)$$

Beidseitiges multiplizieren mit B führt zu

$$B\dot{\vec{L}}_B = \dot{\underline{\vec{L}}}_B = B\vec{M}_B = \vec{\underline{M}}_B \quad (4.81)$$

mit der Projektion $\dot{\underline{\vec{L}}}_B$ der zeitlichen Drehimpulsableitung ins Robotersystem aus (4.67) und der Momentensumme $\vec{\underline{M}}_B$ aus (4.74). Die für die Bestimmung der Momentensumme notwendigen Schnittmomente $\vec{\underline{T}}_{Hi}$ und $\vec{\underline{T}}_{Si}$ stehen mit (4.43)-(4.45) bzw. (4.62)-(4.64) und die notwendigen Schnittkräfte $\vec{\underline{A}}_{Hi}$ und $\vec{\underline{A}}_{Si}$ mit (4.40) bzw. (4.59) bereit. Somit ergeben sich die Gleichungen

4. Regelung eines omnidirektionalen Antriebssystems

$$\begin{aligned}
& \underline{\underline{M}}_{Bx} = 0 \\
& = \sum_i \left[\frac{1}{2r_i} \left(-\gamma_i \underline{\underline{J}}_{Siox} \left(\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i) \right) - \underline{\underline{J}}_{Hiox} \sin(2\varphi_i) \right) \right] \underline{\underline{\ddot{r}}}_x \\
& + \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{Siox} \left(\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i) \right) + \underline{\underline{J}}_{Hiox} \left(\cos(2\varphi_i) + 1 \right) \right) \right. \\
& \quad \left. + m_{Hi}(\underline{\underline{h}}_{iz} - \underline{\underline{b}}_z) + m_{Si}(\underline{\underline{s}}_{iz} - \underline{\underline{b}}_z) \right] \underline{\underline{\ddot{r}}}_y \\
& + \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{Siox} \left(\underline{\underline{h}}_{iy} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) \right. \right. \right. \\
& \quad \left. \left. + \underline{\underline{h}}_{ix} (\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i)) \right) \right. \\
& \quad \left. + \underline{\underline{J}}_{Hiox} \left(\underline{\underline{h}}_{iy} \sin(2\varphi_i) + \underline{\underline{h}}_{ix} (\cos(2\varphi_i) + 1) \right) \right) \\
& \quad \left. + m_{Hi} \underline{\underline{h}}_{ix} (\underline{\underline{h}}_{iz} - \underline{\underline{b}}_z) + m_{Si} \underline{\underline{s}}_{ix} (\underline{\underline{s}}_{iz} - \underline{\underline{b}}_z) \right] \underline{\underline{\ddot{\Psi}}} \\
& + \sum_i \left[\frac{1}{r_i} \left(-\gamma_i \underline{\underline{J}}_{Siox} \cos(\varphi_i + \beta_i) - \underline{\underline{J}}_{Hiox} \cos(2\varphi_i) \right) \right] \underline{\underline{\dot{r}}}_x \underline{\underline{\dot{\Psi}}} \\
& + \sum_i \left[\frac{1}{r_i} \left(-\gamma_i \underline{\underline{J}}_{Siox} \sin(\varphi_i + \beta_i) - \underline{\underline{J}}_{Hiox} \sin(2\varphi_i) \right) \right] \underline{\underline{\dot{r}}}_y \underline{\underline{\dot{\Psi}}} \\
& + \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{Siox} \left(\underline{\underline{h}}_{iy} (\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i)) \right. \right. \right. \\
& \quad \left. \left. + \underline{\underline{h}}_{ix} (\sin(\varphi_i - \beta_i) - \sin(\varphi_i + \beta_i)) \right) \right. \\
& \quad \left. + \underline{\underline{J}}_{Hiox} \left(\underline{\underline{h}}_{iy} (\cos(2\varphi_i) - 1) - \underline{\underline{h}}_{ix} \sin(2\varphi_i) \right) \right) \\
& \quad \left. + m_{Hi} \underline{\underline{h}}_{iy} (\underline{\underline{b}}_z - \underline{\underline{h}}_{iz}) + m_{Si} \underline{\underline{s}}_{iy} (\underline{\underline{b}}_z - \underline{\underline{s}}_{iz}) \right] \underline{\underline{\dot{\Psi}}}^2 \\
& + \sum_i \left[m_{Hi} (\underline{\underline{b}}_y - \underline{\underline{h}}_{iy}) + m_{Si} (\underline{\underline{b}}_y - \underline{\underline{s}}_{iy}) \right] g \\
& + \sum_i \left[\left(k_i \sin(\varphi_i) (\underline{\underline{b}}_z - \underline{\underline{h}}_{iz} + r_i) + \underline{\underline{h}}_{iy} - \underline{\underline{b}}_y \right) Z_i \right] + \sum_i \left[\cos(\varphi_i) \left(\underline{\underline{b}}_z - \underline{\underline{h}}_{iz} + r_i \right) D_i \right],
\end{aligned} \tag{4.82}$$

4.1. Modellierung eines omnidirektional getriebenen Roboters

$$\begin{aligned}
& \underline{M}_y = 0 \\
& = \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \left(\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i) \right) + \underline{\underline{J}}_{H_{ixx}} \left(\cos(2\varphi_i) - 1 \right) \right) \right. \\
& \quad \left. + m_{Hi}(\underline{b}_z - \underline{h}_{iz}) + m_{Si}(\underline{b}_z - \underline{s}_{iz}) \right] \underline{\underline{\ddot{r}}}_x \\
& + \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \left(\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i) \right) + \underline{\underline{J}}_{H_{ixx}} \sin(2\varphi_i) \right) \right] \underline{\underline{\ddot{r}}}_y \\
& + \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \left(\underline{h}_{iy}(\cos(\varphi_i - \beta_i) - \cos(\varphi_i + \beta_i)) \right. \right. \right. \\
& \quad \left. \left. + \underline{h}_{ix}(\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i)) \right) \right. \\
& \quad \left. + \underline{\underline{J}}_{H_{ixx}} \left(\underline{h}_{iy}(1 - \cos(2\varphi_i)) + \underline{h}_{ix} \sin(2\varphi_i) \right) \right) \\
& \quad \left. + m_{Hi} \underline{h}_{iy}(\underline{h}_{iz} - \underline{b}_z) + m_{Si} \underline{s}_{iy}(\underline{s}_{iz} - \underline{b}_z) \right] \underline{\underline{\ddot{\Psi}}} \\
& + \sum_i \left[\frac{1}{r_i} \left(-\gamma_i \underline{\underline{J}}_{S_{ixx}} \sin(\varphi_i + \beta_i) - \underline{\underline{J}}_{H_{ixx}} \sin(2\varphi_i) \right) \right] \underline{\underline{\dot{r}}}_x \underline{\underline{\dot{\Psi}}} \\
& + \sum_i \left[\frac{1}{r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \cos(\varphi_i + \beta_i) + \underline{\underline{J}}_{H_{ixx}} \cos(2\varphi_i) \right) \right] \underline{\underline{\dot{r}}}_y \underline{\underline{\dot{\Psi}}} \\
& + \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \left(\underline{h}_{iy}(\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) \right. \right. \right. \\
& \quad \left. \left. + \underline{h}_{ix}(\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i)) \right) \right. \\
& \quad \left. + \underline{\underline{J}}_{H_{ixx}} \left(\underline{h}_{iy} \sin(2\varphi_i) + \underline{h}_{ix}(\cos(2\varphi_i) + 1) \right) \right) \\
& \quad \left. + m_{Hi} \underline{h}_{ix}(\underline{h}_{iz} - \underline{b}_z) + m_{Si} \underline{s}_{ix}(\underline{s}_{iz} - \underline{b}_z) \right] \underline{\underline{\dot{\Psi}}}^2 \\
& + \sum_i \left[m_{Hi}(\underline{h}_{ix} - \underline{b}_x) + m_{Si}(\underline{s}_{ix} - \underline{b}_x) \right] g \\
& + \sum_i \left[\left(\cos(\varphi_i) k_i (\underline{h}_{iz} - \underline{b}_z - r_i) + \underline{b}_x - \underline{h}_{ix} \right) Z_i \right] + \sum_i \left[\sin(\varphi_i) (\underline{b}_z - \underline{h}_{iz} + r_i) D_i \right]
\end{aligned} \tag{4.83}$$

4. Regelung eines omnidirektionalen Antriebssystems

und

$$\begin{aligned}
 \underline{\underline{M}}_z = \underline{\underline{J}}_{Bzz} \ddot{\Psi} = & \sum_i \left[m_{Hi} (\underline{h}_{iy} - \underline{b}_y) + m_{Si} (\underline{s}_{iy} - \underline{b}_y) \right] \ddot{\underline{r}}_x \\
 & + \sum_i \left[m_{Hi} (\underline{b}_x - \underline{h}_{ix}) + m_{Si} (\underline{b}_x - \underline{s}_{ix}) \right] \ddot{\underline{r}}_y \\
 & + \sum_i \left[m_{Hi} \left(\underline{h}_{iy} (\underline{b}_y - \underline{h}_{iy}) + \underline{h}_{ix} (\underline{b}_x - \underline{h}_{ix}) \right) \right. \\
 & \left. + m_{Si} \left(\underline{s}_{iy} (\underline{b}_y - \underline{s}_{iy}) + \underline{s}_{ix} (\underline{b}_x - \underline{s}_{ix}) \right) - \underline{\underline{J}}_{Hizz} - \underline{\underline{J}}_{Sizz} \right] \ddot{\Psi} \\
 & + \sum_i \left[m_{Hi} (\underline{b}_y \underline{h}_{ix} - \underline{b}_x \underline{h}_{iy}) + m_{Si} (\underline{b}_y \underline{s}_{ix} - \underline{b}_x \underline{s}_{iy}) \right] \dot{\Psi}^2 \\
 & + \sum_i \left[k_i \left(\cos(\varphi_i) (\underline{b}_y - \underline{h}_{iy}) + \sin(\varphi_i) (\underline{h}_{ix} - \underline{b}_x) \right) Z_i \right] \\
 & + \sum_i \left[\left(\sin(\varphi_i) (\underline{h}_{iy} - \underline{b}_y) + \cos(\varphi_i) (\underline{h}_{ix} - \underline{b}_x) \right) D_i \right] \quad (4.84)
 \end{aligned}$$

für den Grundkörper aus dem Drehimpulssatz.

4.2. Beschränkung der Beschleunigungsfähigkeit eines omnidirektionalen Antriebssystems

In diesem Abschnitt sollen die bisher dargelegten Zusammenhänge durch numerische Auswertung zur Bestimmung der zustandsabhängigen Beschleunigungsfähigkeit eines omnidirektional angetriebenen Roboters nutzbar gemacht werden, welche durch die Haftreibungsgrenze (4.32) der einzelnen Räder und durch die elektrisch realisierbaren Antriebsmomente (4.97) begrenzt wird.

4.2.1. Umstrukturierung von Impuls- und Drehimpulssatz des Grundkörpers

Zur numerischen Auswertung werden die x- und y-Komponente des Impulssatzes sowie die z-Komponente des Drehimpulssatzes (Gleichungen (4.77), (4.78) und (4.84)) zum Gleichungssystem

$$\underline{\ddot{\mathbf{B}}}_{\underline{\underline{\mathbf{Q}}}} = \underline{\mathbf{C}} \underline{\dot{\mathbf{Q}}} + \underline{\mathbf{Z}} \underline{\ddot{\mathbf{z}}} + \underline{\mathbf{D}} \underline{\ddot{\mathbf{d}}} \quad (4.85)$$

und die x- und y-Komponente des Drehimpulssatzes sowie die z-Komponente des Impulssatzes (Gleichungen (4.82), (4.83) und (4.79)) zum Gleichungssystem

$$\underline{\ddot{\mathbf{G}}} = \underline{\mathbf{N}} \underline{\ddot{\mathbf{q}}} + \underline{\mathbf{P}} \underline{\dot{\mathbf{Q}}} + \underline{\mathbf{R}} \underline{\ddot{\mathbf{z}}} + \underline{\mathbf{T}} \underline{\ddot{\mathbf{d}}} \quad (4.86)$$

umgeordnet.

Das Gleichungssystem (4.85) geht dabei aus den Bestimmungsgleichungen der zu beeinflussenden Größen $\ddot{\underline{\underline{\mathbf{x}}}}$, $\ddot{\underline{\underline{\mathbf{y}}}}$ und $\ddot{\underline{\underline{\Psi}}}$ hervor, während sich das Gleichungssystem (4.86) aus den Gleichungen der Beschreibung der Unterdrückung von Nick-, Roll- und Hubbewegungen zusammensetzt. Die in den Gleichungssystemen (4.85) und (4.86) auftretenden Vektoren sind die in das Robotersystem projizierte gemischte Beschleunigung $\underline{\ddot{\mathbf{q}}}$ nach Gleichung 4.10, der gemischte quadratische Geschwindigkeitsvektor $\underline{\dot{\mathbf{Q}}}$ nach Gleichung 4.11 sowie der gemischte Vektor $\underline{\ddot{\mathbf{G}}}$ des Einflusses der Erdbeschleunigung auf die fixierten Koordinaten. Weiterhin treten der Vektor der Haftreibungskraftkomponenten in den jeweiligen Radrichtungen

$$\underline{\ddot{\mathbf{d}}} = (\underline{D}_1 \quad \dots \quad \underline{D}_n)^T \quad (4.87)$$

und der Vektor der Zwangskraftkomponenten zwischen Boden und den Rädern in $\underline{\vec{e}}_z$

$$\underline{\ddot{\mathbf{z}}} = (\underline{Z}_1 \quad \dots \quad \underline{Z}_n)^T \quad (4.88)$$

auf.

	Trägheit	Corioliskräfte	Zwangskräfte	Antriebskräfte
Gesteuerte Koordinaten	$\underline{\mathbf{B}}$	$\underline{\mathbf{C}}$	$\underline{\mathbf{Z}}$	$\underline{\mathbf{D}}$
Fixierte Koordinaten	$\underline{\mathbf{N}}$	$\underline{\mathbf{P}}$	$\underline{\mathbf{R}}$	$\underline{\mathbf{T}}$

Tabelle 4.1.: Verallgemeinerte Matrizen des Mechanikmodells

Die Matrizen sowie der Vektor $\underline{\ddot{\mathbf{G}}}$ sind im Anhang B dargelegt, ihre Bedeutung ist in Tabelle 4.1 zusammengefasst. Dieses Vorgehen hat den Nachteil, dass sich die

4. Regelung eines omnidirektionalen Antriebssystems

verallgemeinerten Matrizen einer direkten physikalischen Deutung entziehen, da in den resultierenden Vektoren Kräfte und Momente gemischt vorliegen. Demgegenüber steht der Vorteil effizienter Auswertbarkeit der sich ergebenden Gleichungssysteme.

Die Matrizen \mathbf{B} , \mathbf{C} , \mathbf{D} , \mathbf{N} , \mathbf{P} und \mathbf{T} sind Konstanten des Roboters. Die Matrizen \mathbf{Z} und \mathbf{R} sind über (4.33) bzw. (4.34) abhängig von der Robotergeschwindigkeit, da sie über die Faktoren k_i Informationen über die Rollreibung der Rollen der Omnidirektionalräder kodieren.

4.2.2. Projektionen der Radwinkelgeschwindigkeiten, der Radgeschwindigkeiten und der Schnittmomente

Für die Bestimmung der Bewegungsinduktionsspannung der Antriebe werden die Radwinkelgeschwindigkeiten der einzelnen Omnidirektionalräder aus (4.18) aufgestellt zu

$$\underbrace{(\omega_1 \quad \dots \quad \omega_n)^T}_{=\vec{\omega}} = \mathbf{W} \dot{\underline{\underline{\mathbf{q}}}}, \quad (4.89)$$

Die Komponenten des Vektors $\vec{\omega}$ enthalten dabei die Winkelgeschwindigkeiten der Omnidirektionalräder um ihre jeweiligen Rotationsachsen $\vec{\underline{\underline{E}}}_{Hix}$.

Für die Bestimmung der Rollreibung der Rollen der Omnidirektionalräder nach Gleichung (4.34) werden die Geschwindigkeitskomponenten $\dot{\underline{\underline{h}}}_{ix} = \vec{\underline{\underline{E}}}_{Hix} \dot{\underline{\underline{h}}}_i$ der einzelnen Räder entlang der jeweiligen Achsrichtungen benötigt. Diese ergeben sich als Komponenten des Vektors

$$\left(\dot{\underline{\underline{h}}}_{1x} \quad \dots \quad \dot{\underline{\underline{h}}}_{nx} \right)^T = \mathbf{X} \dot{\underline{\underline{\mathbf{q}}}}. \quad (4.90)$$

Aus (4.43)-(4.45) und (4.62)-(4.64) ergibt sich der Vektor der Summen $\vec{\underline{\underline{T}}}_{Hi} \vec{\underline{\underline{E}}}_{Hix} + \vec{\underline{\underline{T}}}_{Si} \vec{\underline{\underline{E}}}_{Six}$, also der auf die jeweiligen $\vec{\underline{\underline{E}}}_{Hix}$ bzw. $\vec{\underline{\underline{E}}}_{Six}$ projizierten Schnittmomente $\vec{\underline{\underline{T}}}_{Hi}$ bzw. $\vec{\underline{\underline{T}}}_{Si}$ zu

$$\begin{pmatrix} \vec{\underline{\underline{T}}}_{H1x} + \vec{\underline{\underline{T}}}_{S1x} \\ \vdots \\ \vec{\underline{\underline{T}}}_{Hnx} + \vec{\underline{\underline{T}}}_{Snx} \end{pmatrix} = \mathbf{U} \ddot{\underline{\underline{\mathbf{q}}}} + \mathbf{V} \dot{\underline{\underline{\mathbf{Q}}}} + \text{diag}(r_1, \dots, r_n) \vec{\mathbf{d}}. \quad (4.91)$$

Die Matrizen sind im Anhang B dargelegt.

4.2.3. Approximation der Haftreibungsgrenze

Wird Gleichung (4.86) zu

$$\mathbf{R}\ddot{\mathbf{z}} = \ddot{\mathbf{G}} - \mathbf{N}\ddot{\mathbf{q}} - \mathbf{P}\ddot{\mathbf{Q}} - \mathbf{T}\ddot{\mathbf{d}} \quad (4.92)$$

umgestellt, so wird die Bestimmung der Zwangskräfte für invertierbare \mathbf{R} möglich. Für Roboter mit mehr als 3 Rädern, ist \mathbf{R} jedoch eine Rechteckmatrix und somit nicht invertierbar. Für solche Roboter wird die Pseudoinverse \mathbf{R}^+ von \mathbf{R} herangezogen.^{4 5} Wird nun

$$\ddot{\mathbf{z}}^* = \mathbf{R}^+(\ddot{\mathbf{G}} - \mathbf{N}\ddot{\mathbf{q}} - \mathbf{P}\ddot{\mathbf{Q}} - \mathbf{T}\ddot{\mathbf{d}}) \quad (4.93)$$

gebildet, so ist $\ddot{\mathbf{z}}^*$ der die Gleichung (4.92) erfüllende Vektor der Zwangskräfte, für den $\|\ddot{\mathbf{z}}^*\|_2 \rightarrow \min$. Der Vektor $\ddot{\mathbf{z}}^*$ erfüllt also die Forderungen, die sich aus dem Sperren von Nick-, Roll- und Hubbewegungen aus Impuls- und Drehimpulssatz ergeben, mit kleinstmöglicher Summe der Komponentenquadrate. Der Vektor $\ddot{\mathbf{z}}^*$ wird somit als plausible Näherung zur Bestimmung der Zwangskräfte angesehen, da eine Überschätzung der zu bestimmenden Haftreibungsgrenze durch die minimale Summe der Komponentenquadrate unwahrscheinlich erscheint – durch die Verwendung dieser Lösung kann vermieden werden, Elastizitäten der Bodenkontakte und eine dementsprechende Erweiterung des Roboterzustandes um Nicken, Rollen und Hub vorzunehmen. Die Zwangskräfte $\ddot{\mathbf{z}}^*$ werden in (4.85) eingesetzt:

$$(\mathbf{Z}\mathbf{R}^+\mathbf{N} + \mathbf{B})\ddot{\mathbf{q}} + (\mathbf{Z}\mathbf{R}^+\mathbf{T} - \mathbf{D})\ddot{\mathbf{d}} = (\mathbf{C} - \mathbf{Z}\mathbf{R}^+\mathbf{P})\ddot{\mathbf{Q}} + \mathbf{Z}\mathbf{R}^+\ddot{\mathbf{G}} \quad (4.94)$$

Einsetzen von (4.94) und (4.93) in die Haftreibungsgrenze (4.32) führt mit

$$\begin{aligned} (\mathbf{R}^+\mathbf{N}\ddot{\mathbf{q}} + \mathbf{R}^+\mathbf{T}\ddot{\mathbf{d}})_i \pm \frac{1}{\mu_i} D_i &\leq (\mathbf{R}^+\mathbf{G} - \mathbf{R}^+\mathbf{P}\ddot{\mathbf{Q}})_i \\ \iff \left(\mathbf{R}^+\mathbf{N}\ddot{\mathbf{q}} + (\mathbf{R}^+\mathbf{T} \pm \text{diag}(1/\mu_1, \dots, 1/\mu_n))\ddot{\mathbf{d}} \right)_i &\leq (\mathbf{R}^+\mathbf{G} - \mathbf{R}^+\mathbf{P}\ddot{\mathbf{Q}})_i \end{aligned} \quad (4.95)$$

⁴Die Pseudoinverse D^+ einer Matrix D wird durch Singulärwertzerlegung der Matrix D nach [PTVF92] gewonnen, wobei eine Zerlegung von D der Form $D = U\Sigma V^T$ durchgeführt wird. Hierbei sind $D, U \in \mathbb{R}^{M \times N}$, $\Sigma \in \mathbb{R}^{N \times N}$ und $V \in \mathbb{R}^{N \times N}$, Σ ist die Diagonalmatrix der Singulärwerte, U und V sind Matrizen orthonormaler Spalten. Mit dieser Zerlegung wird die Pseudoinverse zu $D^+ := V[\text{diag}(1/\sigma_j)]U^T$ dargestellt.

⁵Da $D^{-1} = D^+$ für invertierbare D gilt, wird im folgenden die Pseudoinverse D^+ als Verallgemeinerung der Inversen D^{-1} auch für invertierbare D weiterverwendet.

4. Regelung eines omnidirektionalen Antriebssystems

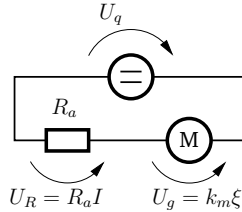


Abbildung 4.7.: Elektrisches Modell

zur positiven bzw. negativen Beschränkung der jeweiligen Antriebskraft. Dabei bezeichnet $(\vec{x})_i$ die Komponente i des Vektors \vec{x} .

4.2.4. Begrenzung durch realisierbare Antriebsmomente

Aus den Maschinengleichungen $U_g = k_m \xi$ und $M = k_m I$ zur Beschreibung der Gleichstrommaschine mit der Winkelgeschwindigkeit ξ um die Maschinenachse, der Ankerstromstärke I und der Maschinenkonstante k_m folgt das von einem Gleichstrommotor am Antrieb bereitgestellte Moment

$$M_d = \frac{k_m}{R_a} U_q - \frac{k_m^2}{R_a} \xi. \quad (4.96)$$

Berücksichtigung der Übersetzung zwischen Antrieb und Rad entsprechend (4.48) führt mit

$$M = \frac{\gamma k_m U_q}{R_a} - \frac{\gamma^2 k_m^2}{R_a} \omega \quad (4.97)$$

zum vom Antrieb am Rad bereitgestellten Moment. Dabei ist R_a der Ankerwiderstand und U_q die Klemmspannung, welche bei Betrieb an einer H-Brückenendstufe der Betriebsspannung U auf $U_q \in [-U_{\max}, U_{\max}]$ begrenzt wird. Das Antriebsmoment wird jedoch nicht nur durch Betriebsspannung, sondern auch durch den realisierbaren Dauerstrom begrenzt:

$$|M| \leq \gamma k_m I_{\max}. \quad (4.98)$$

4.2. Beschränkung der Beschleunigungsfähigkeit

Anwendung von (4.97) und (4.98) für alle Antriebe unter Berücksichtigung von (4.89) führt mit

$$\vec{\mathbf{M}}_Q = \begin{pmatrix} \left(\frac{\gamma k_m U_{\max}}{R_a} \right)_1 \\ \vdots \\ \left(\frac{\gamma k_m U_{\max}}{R_a} \right)_n \end{pmatrix}, \quad (4.99)$$

$$\vec{\mathbf{M}}_G = - \operatorname{diag} \left(\frac{\gamma^2 k_m^2}{R_a} \right) \mathbf{W} \dot{\underline{\underline{\mathbf{q}}}} \quad (4.100)$$

und

$$\vec{\mathbf{M}}_I = \begin{pmatrix} \left(\gamma k_m I_{\max} \right)_1 \\ \vdots \\ \left(\gamma k_m I_{\max} \right)_n \end{pmatrix} \quad (4.101)$$

zu den führt zu den Extremalmomenten

$$\vec{\mathbf{M}}_{\min} = \begin{pmatrix} \max \left(\left(-\vec{\mathbf{M}}_Q + \vec{\mathbf{M}}_G, -\vec{\mathbf{M}}_I \right)_1 \right) \\ \vdots \\ \max \left(\left(-\vec{\mathbf{M}}_Q + \vec{\mathbf{M}}_G, -\vec{\mathbf{M}}_I \right)_n \right) \end{pmatrix} \quad (4.102)$$

und

$$\vec{\mathbf{M}}_{\max} = \begin{pmatrix} \min \left(\left(\vec{\mathbf{M}}_Q + \vec{\mathbf{M}}_G, \vec{\mathbf{M}}_I \right)_1 \right) \\ \vdots \\ \min \left(\left(\vec{\mathbf{M}}_Q + \vec{\mathbf{M}}_G, \vec{\mathbf{M}}_I \right)_n \right) \end{pmatrix}. \quad (4.103)$$

Hieraus folgt aus (4.91)

$$\begin{aligned} \left(\vec{\mathbf{M}}_{\min} \right)_i &\leq - \left(\tilde{T}_i \vec{E}_{xi} + \tilde{T}_{di} \vec{E}_{dxi} \right) \\ &= - \left(\mathbf{U} \dot{\underline{\underline{\mathbf{q}}}} + \mathbf{v} \dot{\underline{\underline{\mathbf{Q}}}} + \operatorname{diag}(r_1, \dots, r_n) \dot{\underline{\underline{\mathbf{d}}}} \right)_i \leq \left(\vec{\mathbf{M}}_{\max} \right)_i \\ \iff \left(\vec{\mathbf{M}}_{\min} + \mathbf{v} \dot{\underline{\underline{\mathbf{Q}}}} \right)_i &\leq - \left(\mathbf{U} \dot{\underline{\underline{\mathbf{q}}}} + \operatorname{diag}(r_1, \dots, r_n) \dot{\underline{\underline{\mathbf{d}}}} \right)_i \leq \left(\vec{\mathbf{M}}_{\max} + \mathbf{v} \dot{\underline{\underline{\mathbf{Q}}}} \right)_i, \end{aligned} \quad (4.104)$$

4. Regelung eines omnidirektionalen Antriebssystems

wobei die projizierten Schnittmomente $\vec{T}_{\underline{i}} \vec{E}_{xi}$ und $\vec{T}_{di} \vec{E}_{dxi}$ aus (4.91) entsprechend der Definition in Abschnitt 4.1.3 negativ angesetzt werden müssen.

4.2.5. Formulierung der Beschleunigungsfähigkeit als lineares Optimierungsproblem

Bisher wurden die sich aus Impuls- und Drehimpulssatz ergebenden vom System zu erfüllenden Gleichungen (4.85) und (4.86) sowie die sich aus den Beschränkungen ergebenden Ungleichungen (4.95) und (4.104) dargelegt. Für die Ermittlung der Maximalbeschleunigung wird eine Skalierungsvariable λ zur richtungstreuen Anpassung einer vorgegebenen Beschleunigung $\ddot{\underline{q}}_r$ zu

$$\ddot{\underline{q}} := \lambda \ddot{\underline{q}}_r \quad (4.105)$$

eingeführt. Hierdurch stellt sich die Beschreibung der Grenzbeschleunigung als lineares Optimierungsproblem der Güte Z dar:

$$Z = \lambda(\ddot{\underline{q}}, \ddot{\underline{q}}_r) \rightarrow \max. \quad (4.106)$$

Hierzu wird das lineare Optimierungsproblem in die Normalform überführt. Der erste Schritt ist die Bereitstellung als Tableau in dem die das System beschreibenden Gleichungen (4.85) und (4.86) sowie die Ungleichungen (4.95) und (4.104) durch Linearkombinationen der Basisvariablen λ, D_1, \dots, D_n auf der linken und die zugehörigen Werte auf der rechten Seite angeordnet werden, siehe Tabelle 4.2.

4.2. Beschränkung der Beschleunigungsfähigkeit

1λ	$+$	$0D_1$	$+\dots+$	$0D_n$	$=$	Z
$(\mathbf{R}^+\mathbf{N}\ddot{\mathbf{q}}_r)_{1\lambda}$	$+$	$((\mathbf{R}^+\mathbf{T})_{1,1} + \frac{1}{\mu_1})D_1$	$+\dots+$	$(\mathbf{R}^+\mathbf{T})_{1,n}D_n$	\leq	$(\mathbf{R}^+\mathbf{G} - \mathbf{R}^+\mathbf{P}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_1}$
					\vdots	
$(\mathbf{R}^+\mathbf{N}\ddot{\mathbf{q}}_r)_{n\lambda}$	$+$	$(\mathbf{R}^+\mathbf{T})_{n,1}D_1$	$+\dots+$	$((\mathbf{R}^+\mathbf{T})_{n,n} + \frac{1}{\mu_n})D_n$	\leq	$(\mathbf{R}^+\mathbf{G} - \mathbf{R}^+\mathbf{P}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_n}$
$(\mathbf{R}^+\mathbf{N}\ddot{\mathbf{q}}_r)_{1\lambda}$	$+$	$((\mathbf{R}^+\mathbf{T})_{1,1} - \frac{1}{\mu_1})D_1$	$+\dots+$	$(\mathbf{R}^+\mathbf{T})_{1,n}D_n$	\leq	$(\mathbf{R}^+\mathbf{G} - \mathbf{R}^+\mathbf{P}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_1}$
					\vdots	
$(\mathbf{R}^+\mathbf{N}\ddot{\mathbf{q}}_r)_{n\lambda}$	$+$	$(\mathbf{R}^+\mathbf{T})_{n,1}D_1$	$+\dots+$	$((\mathbf{R}^+\mathbf{T})_{n,n} - \frac{1}{\mu_n})D_n$	\leq	$(\mathbf{R}^+\mathbf{G} - \mathbf{R}^+\mathbf{P}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_n}$
$-(\mathbf{U}\ddot{\mathbf{q}}_r)_{1\lambda}$	$-$	r_1D_1	$-\dots-$	$0D_n$	\leq	$(\dot{\mathbf{M}}_{\max} + \mathbf{V}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_1}$
					\vdots	
$-(\mathbf{U}\ddot{\mathbf{q}}_r)_{n\lambda}$	$-$	$0D_1$	$-\dots-$	r_nD_n	\leq	$(\dot{\mathbf{M}}_{\max} + \mathbf{V}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_n}$
$-(\mathbf{U}\ddot{\mathbf{q}}_r)_{1\lambda}$	$-$	r_1D_1	$-\dots-$	$0D_n$	\geq	$(\dot{\mathbf{M}}_{\min} + \mathbf{V}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_1}$
					\vdots	
$-(\mathbf{U}\ddot{\mathbf{q}}_r)_{n\lambda}$	$-$	$0D_1$	$-\dots-$	r_nD_n	\geq	$(\dot{\mathbf{M}}_{\min} + \mathbf{V}\dot{\mathbf{Q}})_{\dot{\mathbf{Q}}_n}$
$((\mathbf{Z}\mathbf{R}^+\mathbf{N} + \mathbf{B})\ddot{\mathbf{q}}_r)_{1\lambda}$	$+$	$(\mathbf{Z}\mathbf{R}^+\mathbf{T} - \mathbf{D})_{1,1}D_1$	$+\dots+$	$(\mathbf{Z}\mathbf{R}^+\mathbf{T} - \mathbf{D})_{1,n}D_n$	$=$	$((\mathbf{C} - \mathbf{Z}\mathbf{R}^+\mathbf{P})\dot{\mathbf{Q}} + \mathbf{Z}\mathbf{R}^+\dot{\mathbf{G}})_1$
$((\mathbf{Z}\mathbf{R}^+\mathbf{N} + \mathbf{B})\ddot{\mathbf{q}}_r)_{2\lambda}$	$+$	$(\mathbf{Z}\mathbf{R}^+\mathbf{T} - \mathbf{D})_{2,1}D_1$	$+\dots+$	$(\mathbf{Z}\mathbf{R}^+\mathbf{T} - \mathbf{D})_{2,n}D_n$	$=$	$((\mathbf{C} - \mathbf{Z}\mathbf{R}^+\mathbf{P})\dot{\mathbf{Q}} + \mathbf{Z}\mathbf{R}^+\dot{\mathbf{G}})_2$
$((\mathbf{Z}\mathbf{R}^+\mathbf{N} + \mathbf{B})\ddot{\mathbf{q}}_r)_{3\lambda}$	$+$	$(\mathbf{Z}\mathbf{R}^+\mathbf{T} - \mathbf{D})_{3,1}D_1$	$+\dots+$	$(\mathbf{Z}\mathbf{R}^+\mathbf{T} - \mathbf{D})_{3,n}D_n$	$=$	$((\mathbf{C} - \mathbf{Z}\mathbf{R}^+\mathbf{P})\dot{\mathbf{Q}} + \mathbf{Z}\mathbf{R}^+\dot{\mathbf{G}})_3$

Tabelle 4.2.: Bestimmung der Beschleunigungsgrenze als lineares Optimierungsproblem reeller Basisvariablen

4. Regelung eines omnidirektionalen Antriebssystems

λ'	λ''	\vec{d}^T	$\vec{d}^{\prime T}$	
1	-1	0...0	0...0	= Z
$\mathbf{R}^+ \mathbf{N}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$-\mathbf{R}^+ \mathbf{N}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$\mathbf{R}^+ \mathbf{T} + \text{diag}(\frac{1}{\mu_i})$	$-\mathbf{R}^+ \mathbf{T} - \text{diag}(\frac{1}{\mu_i})$	$\begin{matrix} < \\ \vdots \\ > \end{matrix} \mathbf{R}^+ \mathbf{G} - \mathbf{R}^+ \mathbf{P}_{\vec{\mathbf{Q}}}^{\dot{\mathbf{Q}}}$
$\mathbf{R}^+ \mathbf{N}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$-\mathbf{R}^+ \mathbf{N}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$\mathbf{R}^+ \mathbf{T} - \text{diag}(\frac{1}{\mu_i})$	$-\mathbf{R}^+ \mathbf{T} + \text{diag}(\frac{1}{\mu_i})$	$\begin{matrix} < \\ \vdots \\ > \end{matrix} \mathbf{R}^+ \mathbf{G} - \mathbf{R}^+ \mathbf{P}_{\vec{\mathbf{Q}}}^{\dot{\mathbf{Q}}}$
$-\mathbf{U}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$+\mathbf{U}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$-\text{diag}(r_i)$	$+\text{diag}(r_i)$	$\begin{matrix} < \\ \vdots \\ > \end{matrix} \vec{\mathbf{M}}_{\max} + \mathbf{V}_{\vec{\mathbf{Q}}}^{\dot{\mathbf{Q}}}$
$-\mathbf{U}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$+\mathbf{U}_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$-\text{diag}(r_i)$	$+\text{diag}(r_i)$	$\begin{matrix} > \\ \vdots \\ > \end{matrix} \vec{\mathbf{M}}_{\min} + \mathbf{V}_{\vec{\mathbf{Q}}}^{\dot{\mathbf{Q}}}$
$(\mathbf{Z}\mathbf{R}^+ \mathbf{N} + \mathbf{B})_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$-(\mathbf{Z}\mathbf{R}^+ \mathbf{N} + \mathbf{B})_{\vec{\mathbf{q}}_r}^{\ddot{\mathbf{q}}_r}$	$+\mathbf{Z}\mathbf{R}^+ \mathbf{T} - \mathbf{D}$	$-\mathbf{Z}\mathbf{R}^+ \mathbf{T} + \mathbf{D}$	$\begin{matrix} = \\ = \\ = \end{matrix} (\mathbf{C} - \mathbf{Z}\mathbf{R}^+ \mathbf{P})_{\vec{\mathbf{Q}}}^{\dot{\mathbf{Q}}} + \mathbf{Z}\mathbf{R}^+ \vec{\mathbf{G}}$

Tabelle 4.3.: Bestimmung der Beschleunigungsgrenze als lineares Optimierungsproblem nichtnegativer Basisvariablen

<div> <div>Masse des Grundkörpers</div> <div>2 kg</div> </div> <div> <div>Masse eines Rades</div> <div>59 g</div> </div> <div> <div>Masse eines Direktantriebs</div> <div>41 g</div> </div> <div> <div>Schwerpunkt des Grundkörpers bezüglich des Achsschnittpunktes</div> <div>$(0 \quad -11 \quad 19)^T \text{ mm}$</div> </div> <div> <div>Radwinkel</div> <div>$15^\circ, 165^\circ$ und 270°</div> </div> <div> <div>Abstand der Räder vom Achsschnittpunkt</div> <div>80 mm</div> </div> <div> <div>Abstand der Antriebe vom Achsschnittpunkt</div> <div>45 mm</div> </div> <div> <div>Haftreibungszahl</div> <div>0,375</div> </div> <div> <div>Rollreibungszahl</div> <div>0,02</div> </div> <div> <div>Maschinenkonstante</div> <div>51 mNm</div> </div> <div> <div>Ankerwiderstand</div> <div>5 Ω</div> </div>
--

Tabelle 4.4.: Parameter des Modells für den aktuellen Prototypen

4.2. Beschränkung der Beschleunigungsfähigkeit

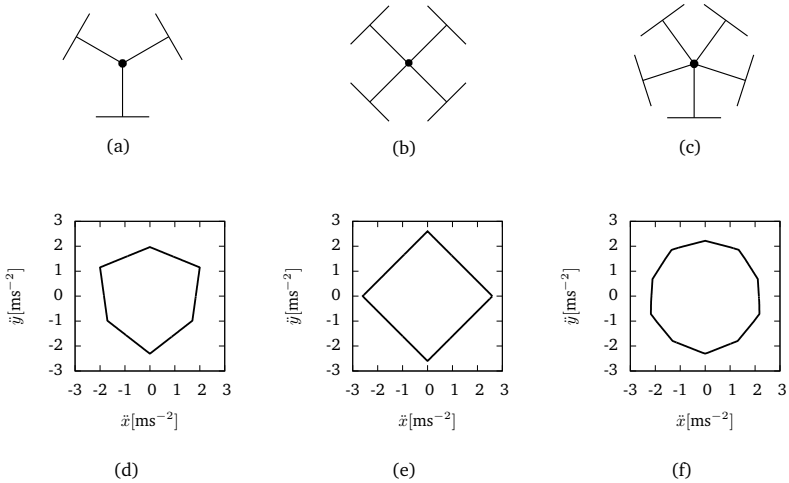


Abbildung 4.8.: Planares Beschleunigungsvermögen im Stillstand (unten) für regelmäßige Antriebskonfigurationen mit Schwerpunkt im Achsschnittpunkt (oben). Realisierbare Beschleunigungen liegen innerhalb der jeweiligen Polygone. Für bessere Vergleichbarkeit der verschiedenen Anordnungen wird der Maximalstrom eines Antriebs der dreirädrigen Konfiguration (a,d) auf $\frac{3}{4}$ bei der vierrädrigen Konfiguration (b,e) und $\frac{3}{5}$ bei der fünfrädrigen Konfiguration (c,f) beschränkt.

4. Regelung eines omnidirektionalen Antriebssystems

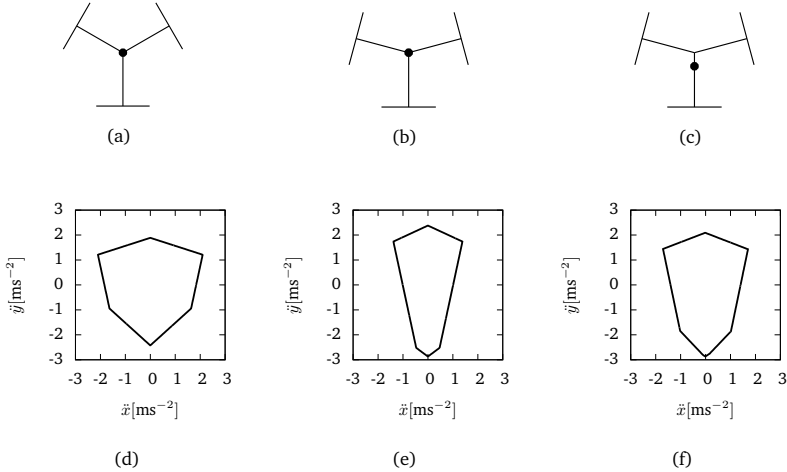


Abbildung 4.9.: Planares Beschleunigungsvermögen im Stillstand (unten) für dreirädrige Antriebskonfigurationen mit Schwerpunkt über dem Achsschnittpunkt (oben). Die regelmäßige Anordnung (a,d) ist schwer mit den konstruktiven Anforderungen an einen RoboCup-Spieler vereinbar. Um einen ausreichend breiten Schussmechanismus aufzunehmen, wird der Winkel zwischen den Vorderachsen auf 150° vergrößert (b,e). Hierdurch verbessert sich die Beschleunigungsfähigkeit des Spielers in Längsrichtung auf Kosten der Querschleunigungsfähigkeit. Durch die Anordnung der Akkumulatoren im Heck des Spielers (vergleiche Abbildung 1.3(c) auf Seite 5) verlagert sich der Schwerpunkt des Grundkörpers nach hinten (c,f). Hierdurch wird der Nachteil der reduzierten Querschleunigungsfähigkeit teilweise kompensiert.

4.2. Beschränkung der Beschleunigungsfähigkeit

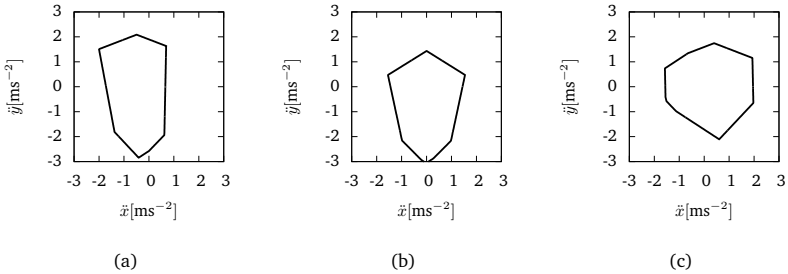


Abbildung 4.10.: Zustandsabhängige Beschleunigungsfähigkeit der Konfiguration nach 4.9(c). Beschleunigungsgrenzen bei schneller Fahrt nach links (a), nach vorn (b) sowie bei schneller Rotation (c).

Mit Hilfe des so formulierten linearen Optimierungsproblems kann eine vorgegebene Beschleunigung $\ddot{\mathbf{q}}_r$ richtungstreu auf das realisierbare Maximum $\ddot{\mathbf{q}}_{\max}$ begrenzt werden. Dafür wird das Simplexverfahren [GRHHM89] der Implementierung [PTVF92] verwendet. Da diese Implementierung bzw. das Simplexverfahren selbst die Nichtnegativität der Basisvariablen voraussetzt, welche bei vorliegendem Problem nicht gegeben ist, werden die Ersetzungen $\lambda =: \lambda' - \lambda''$ und $D_i =: D'_i - D''_i$ vorgenommen und somit das Tableau in Normalform mit nichtnegativen Basisvariablen überführt, siehe Tabelle 4.3. Nach diesem Schritt ist dafür zu sorgen, dass alle rechtsseitigen Koeffizienten des Tableaus nichtnegativ sind. Weiterhin muss das Tableau danach so umgeordnet werden, dass der Zeile der Gütedefinition zuerst alle kleiner-Ungleichungen und danach alle größer-Ungleichungen folgen. Das Ende des Tableaus bilden die Gleichungszeilen. Diese Anforderungen werden durch den Einsatz von Hüllklassen zur Modellierung einer Zeile sowie eines Tableaus erfüllt. Hierdurch wird die sichere Verwendung der Funktion zur linearen Optimierung nach [PTVF92] auf einfache Weise sichergestellt. Nach der Rücksubstitution der nichtnegativen Basisvariablen umfasst das Ergebnis des Simplexverfahrens den maximal unter Einhaltung aller Nebenbedingungen realisierbaren Skalierungsfaktor λ sowie die zugehörige Antriebskraftkonfiguration $\ddot{\mathbf{d}}$. Durch Anwendung der Gleichung (4.91) können hieraus die Antriebsmomente bzw. mit Gleichung (4.97) die Klemmspannungen bestimmt werden.

In der beschriebenen Variante des Moduls zur Bestimmung der Beschleunigungsfähigkeit wird für jede geforderte Sollbeschleunigung $\ddot{\mathbf{q}}_r$ der Verstärkungsfaktor

4. Regelung eines omnidirektionalen Antriebssystems

λ gefunden, für den $\ddot{\vec{q}} = \lambda \ddot{\vec{q}}_r$ auf der Aussteuergrenze liegt. Durch Einfügen der Ungleichung $1\lambda + 0D_1 + \dots 0D_n \leq 1$ in Tabelle 4.2 kann erreicht werden, dass zwar große $\ddot{\vec{q}}_s$ auf das realisierbare Maximum begrenzt, kleine $\ddot{\vec{q}}_r$ jedoch nicht verstärkt werden. Das Modul geht somit in ein Modell der entkoppelnden Beschleunigungsbegrenzung über und wird für den Einsatz in einem Geschwindigkeitsregelkreis nutzbar.

Mit der Definition einer Funktion Λ zu

$$\Lambda : \Omega \times \Omega \rightarrow \mathbb{R}$$

$$(\vec{q}, \dot{\vec{q}}_r) = (\vec{q}^T \quad \dot{\vec{q}}^T \quad \ddot{\vec{q}}_r^T \quad \ddot{\vec{q}}_r^T)^T \mapsto \Lambda(\vec{q}, \dot{\vec{q}}_r) = \lambda(B(\vec{q})\dot{\vec{q}}, B(\vec{q})\ddot{\vec{q}}_r) \quad (4.107)$$

sei die Bestimmung der maximalen Beschleunigungsskalierung durch das Simplexverfahren für Tabelle 4.3 im folgenden bezeichnet. Hierbei werden bereits die Projektionen von Roboter Geschwindigkeit und Sollbeschleunigung in das Robotersystem vorgenommen.

In den Abbildungen 4.8-4.10 finden sich Beispiele ermittelter Beschleunigungsgrenzen für verschiedene Antriebskonfigurationen. Abbildung 4.8 zeigt den Einfluss der Radanzahl regelmäßiger Antriebskonfigurationen. In Abbildung 4.9 verdeutlicht den Einfluss von Schwerpunktlage und Radanordnung des aktuellen Prototypen⁶ im Gegensatz zur regelmäßigen Radanordnung mit idealisierter Schwerpunktlage. In Abbildung 4.10 werden Beschleunigungsgrenzen des aktuellen Prototypen bei verschiedenen Geschwindigkeiten dargestellt.

4.2.6. Erweiterung des Modells um den Radschlupf

In Gleichung (4.17) wurde die Rollbedingung für ideales Rollen aufgestellt, um die Rotation der Räder und Antriebe durch die kinematische Verknüpfung mit der Bewegung des Roboters zu beschreiben. Hierdurch ergab sich eine numerisch leicht auswertbare Formulierung für die Bestimmung der Beschleunigungsfähigkeit des omnidirektionalen Antriebssystems. Hierfür wurde der Momentanpol der Abrollbewegung des Rades i idealisiert im Fußpunkt \vec{f}_i (Gleichung (4.14)) angenommen. Somit stand über Gleichung (4.18) die Winkelgeschwindigkeit ω_i des Rades i um seine Radachse zur Verfügung. In der praktischen Anwendung [Lan05, RF06] wird jedoch eine Abweichung zwischen idealisierter und tatsächlicher bzw. durch Messsysteme bestimmter Radwinkelgeschwindigkeit beobachtet. Dieser Effekt wird in [WCGR02] durch ein nichtlineares Modell für omnidirektionale Antriebssysteme beschrieben. Um die

⁶Da dieser Prototyp (siehe Abbildung 1.3 auf Seite 5) momentan noch nicht vollständig ist, werden Schwerpunktlage, Masse und Trägheitsmomente aus den Konstruktionsdaten bestimmt. Haftreibungszahlen werde von den Messungen am Vorgängermodell [Lan05] übernommen. Siehe auch Tabelle 4.4.

4.2. Beschränkung der Beschleunigungsfähigkeit

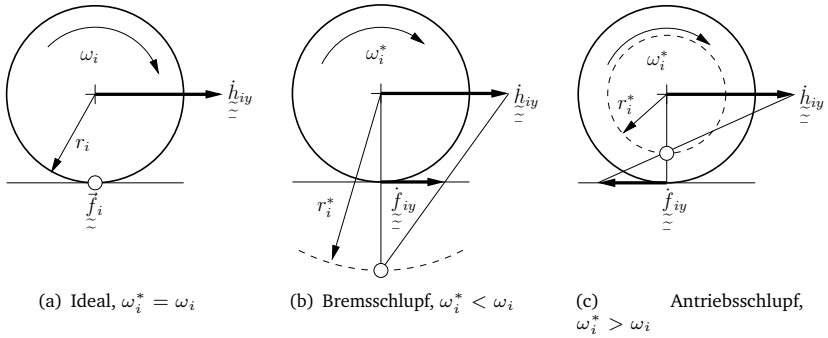


Abbildung 4.11.: Einfluss des Radschlupfs

Auswertung des Dynamikmodells unter Echtzeitanforderungen zu ermöglichen, wird hier vereinfachend eine Korrektur des erarbeiteten Modells um den Schlupf realisiert.

Die Abweichung zwischen idealisierter und tatsächlicher Radwinkelgeschwindigkeit bedeutet, dass der Fußpunkt nicht der Momentanpol der realen Abrollbewegung ist. Statt dessen verlagert sich der Momentanpol durch den bei der Übertragung einer Antriebskraft D_i am Rad i auftretenden Schlupf entlang der Geraden durch Nabenzentrum und Fußpunkt, siehe Abbildung 4.11.

Sei r_i^* der Abstand zwischen Momentanpol und Radnabe und ω_i^* die tatsächliche Winkelgeschwindigkeit des Rades um seine Achse, so wird Gleichung (4.18) präzisiert zu

$$\omega_i^* = \frac{1}{r_i^*} \left(\sin(\varphi_i) \underline{\dot{x}} - \cos(\varphi_i) \underline{\dot{y}} - (\cos(\varphi_i) \underline{h}_{ix} + \sin(\varphi_i) \underline{h}_{iy}) \underline{\Psi} \right) = \frac{r_i}{r_i^*} \omega_i. \quad (4.108)$$

Hierbei stehen an einem realen System ω_i^* durch Messung und ω_i durch die idealisierte Betrachtung entsprechend Gleichung (4.18) bzw. (4.89) bereit. Der Radschlupf äußert sich also in Form einer Übersetzung. Hierdurch wird es möglich, die Gleichungen (4.19) und (4.49) zur Beschreibung des Drehimpulses von Rad bzw. Antrieb mit der Substitution $\omega_i^* = \frac{\omega_i^*}{\omega_i} \omega_i$ zu modifizieren. Damit ergeben sich die präzisierten Drehimpulse durch Multiplikation der Komponenten

4. Regelung eines omnidirektionalen Antriebssystems

L_{Hix} , L_{Hiy} und L_{Six} , L_{Siy} mit den Faktoren $\frac{\omega_i^*}{\omega_i}$.^{7 8} Unter der Annahme stationären Radschlupfs bleiben diese Faktoren bei der Zeitableitung der Drehimpulse erhalten und führen zu einer Modifikation der Matrizen des Gleichungssystems der fixierten Koordinaten. Diese Modifikation soll stellvertretend am Beispiel des Eintrags $n_{1,1}$ der Matrix \mathbf{N} nach Gleichung (B.14) skizziert werden:

$$\begin{aligned}
 n_{1,1} &= \sum_i \left[\frac{1}{2r_i} \left(-\gamma_i \underline{\underline{J}}_{Siox} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) - \underline{\underline{J}}_{Hiox} \sin(2\varphi_i) \right) \right] \\
 &\quad \Downarrow \text{wird präzisiert zu} \\
 n_{1,1} &= \sum_i \left[\frac{\omega_i^*}{\omega_i} \frac{1}{2r_i} \left(-\gamma_i \underline{\underline{J}}_{Siox} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) - \underline{\underline{J}}_{Hiox} \sin(2\varphi_i) \right) \right] \\
 &= - \begin{pmatrix} \frac{\gamma_1 \underline{\underline{J}}_{S1xx}}{\underline{\underline{J}}_{S1xx}} (\sin(\varphi_1 + \beta_1) + \sin(\varphi_1 - \beta_1)) + \frac{\underline{\underline{J}}_{H1xx}}{2r_1} \sin(2\varphi_1) \\ \vdots \\ \frac{\gamma_n \underline{\underline{J}}_{Snxx}}{\underline{\underline{J}}_{Snxx}} (\sin(\varphi_n + \beta_n) + \sin(\varphi_n - \beta_n)) + \frac{\underline{\underline{J}}_{Hnxx}}{2r_n} \sin(2\varphi_n) \end{pmatrix} \begin{pmatrix} \frac{\omega_1^*}{\omega_1} \\ \vdots \\ \frac{\omega_n^*}{\omega_n} \end{pmatrix}. \quad (4.109)
 \end{aligned}$$

Weiterhin müssen in gleicher Weise die Matrizen \mathbf{U} und \mathbf{V} aus Gleichung (4.91) angepasst werden. Zur Anpassung der elektrischen Begrenzung wird das aus der Induktionsspannung resultierende Moment in Gleichung (4.101) nicht aus der Robotergeschwindigkeit, sondern aus den gemessenen Radwinkelgeschwindigkeiten bestimmt.

Für die Annahme stationären Radschlupfs sind diese Anpassungen sehr gering und beeinflussen den Zeitaufwand für die Auswertung nur unwesentlich. Würde diese Annahme im Sinne höherer Modelltreue jedoch fallen gelassen, so ergäbe sich ein sehr viel komplexeres und nichtlineares System mit beträchtlichem Mehraufwand für die Auswertung. Die Hauptaufgabe des Modells ist jedoch der Einsatz im Betrieb eines Reglers, deshalb erscheint diese Annahme bzw. Einschränkung als gerechtfertigt.

⁷ Für $\omega_i = 0$ weist das präzisierte Modell eine Singularität auf. Diese Singularität ist eine direkte Folge aus der Modellannahme der kinematischen Verknüpfung von Robotergeschwindigkeit und Radwinkelgeschwindigkeiten. Es erscheint als zweckmäßig, den Radschlupf an dieser Stelle nicht zu beachten.

⁸ Aus Gründen der Übersichtlichkeit werden die Trägheitsmomente $\underline{\underline{J}}_{Hiox}$ und $\underline{\underline{J}}_{Siox}$ durch ihre jeweiligen Produkte mit $\frac{\omega_i^*}{\omega_i}$ substituiert. Hierdurch wird die konsistente Beeinflussung aller von den Drehimpulskomponenten L_{Hix} , L_{Hiy} und L_{Six} , L_{Siy} abhängigen Gleichungen sichergestellt.

4.2.7. Vereinfachung für dreirädrige Roboter

Die Bestimmung der Beschleunigungsfähigkeit eines omnidirektional angetriebenen Roboters beliebiger Radanzahl wird in Abschnitt 4.2.5 als lineares Optimierungsproblem beschrieben und durch das Simplexverfahren gelöst. Das Simplexverfahren findet dabei die optimale Lösung der durch ein Tableau beschriebenen Optimierungsaufgabe. In diesem Tableau sind die Beschränkungen durch elektrisch realisierbare Antriebsmomente und mechanisch übertragbare Antriebskräfte als Ungleichung und die zu erfüllenden Sätze der Mechanik als Gleichungen dargestellt. Als Basisvariablen treten in diesem Tableau der Skalierungsfaktor λ sowie die Antriebskräfte D_1, \dots, D_n auf. Das Tableau ist weiterhin abhängig von der Sollbeschleunigung und der aktuellen Geschwindigkeit. Die Matrizen \mathbf{R} und \mathbf{Z} nach (B.26) bzw. (B.9) sind über die Beschreibung der Rollreibung der Rollen der Omnidirektionalräder nach (4.33) bzw. (4.34) Abhängige der Geschwindigkeit des Roboters.

Für dreirädrige Roboter umfasst das Tableau also 4 Basisvariablen, 3 Gleichungen und 12 Ungleichungen. Durch Einsetzen der Gleichungen kann es umgeformt werden in ein Ersatztableau, welches 12 Ungleichungen in Abhängigkeit einer Basisvariablen, dem Skalierungsfaktor λ enthält.

Wird weiterhin das Rollreibungsgesetz nach (4.33) zu

$$H_i = -\mu_i \left(l \left(\dot{\vec{h}}_i \vec{E}_{Hix} \right) - 1 \right) Z_i = k_i Z_i, \quad l(v) := \begin{cases} 0 & -v > \varepsilon \\ 2 & v > \varepsilon \\ 1 & \text{sonst} \end{cases} \quad (4.110)$$

modifiziert, so wird die Betrachtung des Systems in $27 = 3^3$ diskreten Betriebsfällen ermöglicht. Somit können wesentliche Anteile des Tableaus vorberechnet werden, wodurch ein erheblicher Vorteil bezüglich der Ausführungsgeschwindigkeit des Verfahrens erwartet wird. Der geschwindigkeitsabhängige Betriebsfall j kann dann zu

$$j := 3^2 l \left(\left(\mathbf{x}_{\underline{\dot{q}}} \right)_1 \right) + 3^1 l \left(\left(\mathbf{x}_{\underline{\dot{q}}} \right)_2 \right) + 3^0 l \left(\left(\mathbf{x}_{\underline{\dot{q}}} \right)_3 \right) + 1 \quad (4.111)$$

bestimmt werden.

Durch einfache Umformungen der Gleichungen (4.85) und (4.86) können die Ungleichungen (4.95) und (4.104) in das Ersatztableau

$$\lambda \left(\mathfrak{A}_{j, \underline{\dot{q}}} \right)_k \leq \left(\mathfrak{B}_j + \mathfrak{D} + \mathfrak{C}_j \underline{\dot{Q}} \right)_k \quad (4.112)$$

4. Regelung eines omnidirektionalen Antriebssystems

überführt werden. Hierbei beschreibt der Term \mathfrak{A}_j den Einfluss der Sollbeschleunigung $\ddot{\underline{\mathbf{q}}}$ auf das Tableau im Betriebsfall j . Der Term \mathfrak{B}_j ist eine Konstante des Betriebsfalles. Weiterhin werden im Term $\mathfrak{D} = \mathfrak{D}(\dot{\underline{\mathbf{q}}})$ die geschwindigkeitsabhängigen elektrisch realisierbaren Momente zusammengefasst und der Term \mathfrak{C}_j stellt den Einfluss der Corioliskräfte dar. Die Matrizen finden sich in Anhang C. Das Tableau (4.112) kann zeilenweise zur Bestimmung des Wertebereiches der Skalierungsvariable ausgewertet und somit das maximale λ bestimmt werden. Wurden die wesentlichen Umformungen bereits während der Initialisierungsphase durchgeführt, so reduziert sich die Lösung des Optimierungsproblems erheblich. Bei der Lösung des vereinfachten Optimierungsproblems konnte so ein Geschwindigkeitszuwachs um Faktor⁹ 18 von 2,86 auf 0,16 μs^{10} pro Bestimmung der Skalierungsvariablen erreicht werden. Diese Werte legen nahe, dass das bereitgestellte Modell zur Ermittlung der Beschleunigungsfähigkeit, zumindest in der vereinfachten Formulierung für dreirädrige Anordnungen, zur Antriebsregelung mit verfügbaren eingebetteten informationstechnischen Systemen genutzt werden kann. Hierbei sollte es möglich sein, einen Regler mit einer Taktung im Bereich von $10^2 \dots 3 \text{ Hz}$ zu realisieren.

⁹Hierbei sind die Implementierungen jedoch nicht vergleichbar. Die Lösung durch das Simplexverfahren erfolgte in einem objektorientierten Kontext mit relativ hohem Zusatzaufwand für die Bereitstellung und Bereinigung von Speicher. Diese Variante ist die erste implementierte Version. Bei der Implementierung für das Ersatztableau wurde besonders auf effiziente Verarbeitung Wert gelegt.

¹⁰Intel®Core™2 Duo CPU E6850, 3GHz

4.3. Regelung

4.3.1. Geschwindigkeitsregler

Das Dynamikmodell aus Abschnitt 4.2 ermöglicht es, Geschwindigkeitsfehler des Roboters an der Beschleunigungsgrenze zu korrigieren, siehe Abbildung 4.12. Die Führungsgröße des Geschwindigkeitsreglers ist die Sollgeschwindigkeit $\dot{\mathbf{q}}_r$ im globalen Koordinatensystem und die Regelgröße ist die Geschwindigkeit $\dot{\mathbf{q}}$ im globalen System. Diese steht jedoch nicht direkt, sondern durch das mitbewegte Messsystem in das Robotersystem projiziert zu $\dot{\tilde{\mathbf{q}}}$ zur Verfügung. Um die Regelabweichung $\tilde{\mathbf{e}}$ bezüglich des ortsfesten Koordinatensystems zu bestimmen, muss die Robotergeschwindigkeit durch Drehung in das globale Koordinatensystem ermittelt werden. Die Regelabweichung wird mit einer Verstärkung k_v in eine globale Sollbeschleunigung $\ddot{\mathbf{q}}_r$ überführt. Die globale Sollbeschleunigung wird in das Robotersystem zu $\ddot{\tilde{\mathbf{q}}}_r$ projiziert und zusammen mit $\dot{\tilde{\mathbf{q}}}$ an das Beschleunigungsmodell übergeben. Im Beschleunigungsmodell wird das Optimierungsproblem (4.106) gelöst und damit der maximale Verstärkungsfaktor λ bestimmt, für den die Beschleunigung $\ddot{\mathbf{q}} = \lambda \ddot{\tilde{\mathbf{q}}}_r$ umgesetzt werden kann. Weiterhin liefert das Beschleunigungsmodell die hierzu notwendigen Antriebskräfte $\tilde{\mathbf{d}}$. Durch $\ddot{\mathbf{q}}$, $\tilde{\mathbf{d}}$ und $\dot{\tilde{\mathbf{q}}}$ ist es möglich, die Klemmspannungen, Ankerströme oder Antriebsmomente der einzelnen Antriebe zu bestimmen.

Der Verstärkungsfaktor k_v und die Begrenzung des Skalierungsfaktors $\lambda \leq 1$ bewirken, dass bei einem digitalen Regler mit einer kleinen¹¹ Totzeit ΔT keine Oszillation des Systems um die Sollgeschwindigkeit stattfindet. Ohne diese Maßnahmen würde bei einer quasikontinuierlichen Realisierung des Reglers eine

¹¹Die vorliegenden Simulationen wurden mit Schrittweiten von 1ms und 10ms durchgeführt. Diese Schrittweiten wurden gewählt, da sie den zu erwartenden Totzeiten im realen System entsprechen.

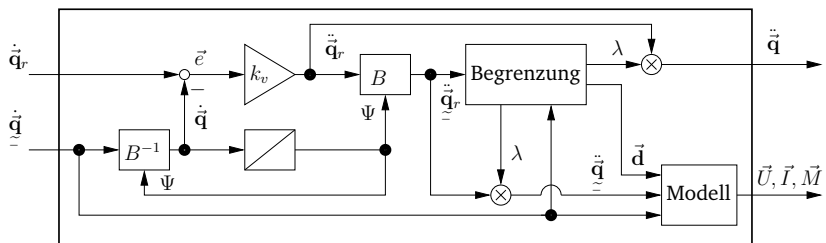


Abbildung 4.12.: Geschwindigkeitsregler

4. Regelung eines omnidirektionalen Antriebssystems

Oszillation der Beschleunigung eintreten, sobald bei kleinen Geschwindigkeitsfehlern die über die Totzeit wirkende Beschleunigung den Geschwindigkeitsfehler übersteigt. Der Parameter k_v wurde zu $k_v := (2\Delta T)^{-1}$ festgelegt. Hierdurch bleibt das Verhalten des Systems bei großen Geschwindigkeitsfehlern unverändert, da k_v keinen Einfluss hat, wenn die mechanischen und elektrischen Begrenzungen greifen. Die Beschleunigungsoszillation bei kleinen Geschwindigkeitsfehlern wird jedoch bedämpft. Antworten auf Führungssprünge des Reglers sind in Abbildung 4.13 dargestellt.

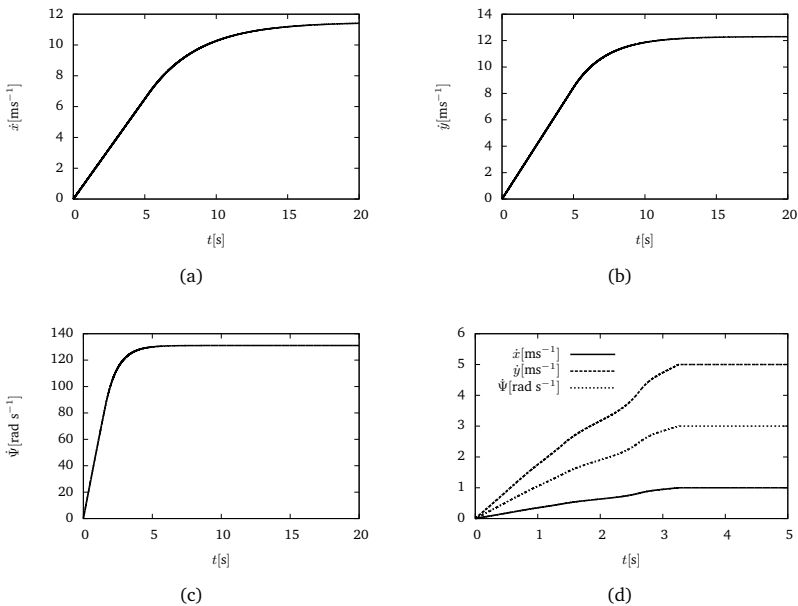


Abbildung 4.13.: Führungssprünge am Geschwindigkeitsregler: Bestimmung der maximalen (a) Quer- und (b) Frontalgeschwindigkeit sowie der maximalen Gierrate (c). Gemischte Bewegung (d).

4.3.2. Lageregler

Der Lageregelkreis hat die Aufgabe, den Roboter in eine Solllage zu überführen. Das Ausregeln der Solllage soll dabei möglichst schnell, aber ohne Überschin-

gen geschehen. Hierbei sollen die zustandsabhängigen Begrenzungen der Dynamik des Systems beachtet werden. Das zugrunde liegende Dynamikmodell ist jedoch so komplex, dass eine Formulierung als Optimalsteuerungsproblem [Bel54, Föl88] für verfügbare eingebettete informationstechnische Systeme nicht erfolgversprechend erscheint.

Deshalb werden hier Regelungsgesetze verwendet, welche mit geringem Aufwand numerisch ausgewertet werden können. Im Folgenden werden dazu zwei realisierte Varianten zur Lageregelung dargelegt. Beide Varianten basieren auf einer parametrisierten Beschreibung des Beschleunigungsverlaufes durch eine Ansatzfunktion, die entsprechenden Parameter werden aus den Randbedingungen gewonnen.¹² Die verwendeten Ansätze sind eine lineare Zeitabhängigkeit der Beschleunigung und eine stückweise konstante Beschleunigung.

Die jeweiligen Regler wurden bisher nur in Simulationen untersucht, Implementierung für einen realen Spieler und Verifikation im realen Einsatz stehen noch aus. Bei einer Implementierung für einen realen Roboter sind hierbei noch zahlreiche Teilprobleme zu lösen. Dabei sind insbesondere die Fragen der Messwertaufbereitung sowie die Einflüsse von Totzeiten von zentraler Bedeutung [LZW08]. Es bleibt offen, wie robust sich diese Regler in der Praxis verhalten werden, welchen Einfluss Abweichungen zwischen Modell und Wirklichkeit haben werden. Eine interessante Alternative zum hier verfolgten Ansatz liegt in den adaptiven Regelungsstrategien [BS09, Beh11], mit denen Systeme unbekannter Parameter robust beeinflusst werden können.

Ansatz einer linearen Zeitabhängigkeit der Beschleunigung

Die Bestimmung eines zeitoptimalen Regelungsgesetzes zur Überführung des Roboters in eine Ziellage wird durch die Zustandsabhängigkeit der Beschleunigungsgrenzen zu einem sehr aufwändigen Problem. Die Komplexität der Aufgabe kann jedoch durch einschränkende Zusatzannahmen soweit reduziert werden, dass eine Lösung mit einem für den Anwendungsfall vertretbaren Aufwand gefunden werden kann. Es wird somit ein Optimum in einem reduzierten Lösungsraum gefunden, dieses ist jedoch nicht Optimum im vollständigen Lösungsraum. Die einschränkende Zusatzannahme ist die Suche nach einer Trajektorie in der Schar der Polynome vom Grad 3, die durch Start- und Ziellage verlaufen. Hiermit ist eine kinematische Beschreibung der Trajektorie gegeben, die auf Validität bezüglich der Dynamik getestet und gegebenenfalls angepasst

¹²Diese Randbedingungen sind in beiden Varianten Ist- und Sollzustand. Da die Regler bisher mit einer Sollgeschwindigkeit von $\vec{0}$ in der Sollage verwendet werden, werden sie als Lageregler bezeichnet. Die für eine Verwendung als Zustandsregler, wie sie beispielsweise für die Realisierung einer Zustellbewegung zu einem bewegten Ball wünschenswert wäre, notwendige Behandlung der Erreichbarkeit des Sollzustandes wird bisher nicht beachtet.

4. Regelung eines omnidirektionalen Antriebssystems

werden kann. Diese Annahme wird durch [CDI97, AJ05] inspiriert, wird hier jedoch in stark vereinfachter Form angewendet.

Für den Zeitverlauf ζ einer beliebigen Koordinate sei die zweite Zeitableitung als lineare Funktion zu

$$\ddot{\zeta}(t) = \ddot{\zeta}t + \ddot{\zeta}_0 \quad (4.113)$$

gegeben. Hieraus ergibt sich

$$\dot{\zeta}(t) = \int_0^t \ddot{\zeta}(\tau) d\tau + \dot{\zeta}_0 = \frac{1}{2}\ddot{\zeta}t^2 + \ddot{\zeta}_0 t + \dot{\zeta}_0 \quad (4.114)$$

als erste Zeitableitung von ζ und schließlich ζ zu

$$\zeta(t) = \int_0^t \dot{\zeta}(\tau) d\tau + \zeta_0 = \frac{1}{6}\ddot{\zeta}t^3 + \frac{1}{2}\ddot{\zeta}_0 t^2 + \dot{\zeta}_0 t + \zeta_0. \quad (4.115)$$

Der Zeitverlauf ζ ist somit ein Polynom vom Grad 3, es ergibt sich ein quadratisches Geschwindigkeitsprofil. Durch die Wahl einer beliebigen Endzeit T_e werden unter Einhaltung der Randbedingungen $\zeta_0, \dot{\zeta}_0, \zeta(T_e) = \zeta_e, \dot{\zeta}(T_e) = \dot{\zeta}_e$ die Parameter $\ddot{\zeta}$ und $\ddot{\zeta}_0$ zu

$$\ddot{\zeta} = 6 \frac{\dot{\zeta}_e - \dot{\zeta}_0}{T_e^2} - 12 \frac{\zeta_e - \zeta_0}{T_e^3} \quad (4.116)$$

und

$$\ddot{\zeta}_0 = 6 \frac{\zeta_e - \zeta_0}{T_e^2} - 2 \frac{\dot{\zeta}_e + 2\dot{\zeta}_0}{T_e} \quad (4.117)$$

bestimmt. Die Roboterlage sei als Zeitverlauf $\vec{q}(t)$ entsprechend (4.115) zu

$$\vec{q}(t) = \begin{pmatrix} \zeta_x(t) & \zeta_y(t) & \zeta_\Psi(t) \end{pmatrix}^T \quad (4.118)$$

dargestellt. Mit Hilfe der Funktion \mathcal{Q} zu

$$\begin{aligned} \mathcal{Q} : \mathfrak{Q} \times \mathfrak{Q} \times \mathbb{R} &\rightarrow \mathfrak{Q}^{\mathbb{R}} \\ (\vec{q}, \vec{q}_r, T_e) &\mapsto \mathcal{Q}(\vec{q}, \vec{q}_r, T_e) = \vec{q}(t) = \begin{pmatrix} \zeta_x(t) & \zeta_y(t) & \zeta_\Psi(t) & \dot{\zeta}_x(t) & \dot{\zeta}_y(t) & \dot{\zeta}_\Psi(t) \end{pmatrix}^T \end{aligned} \quad (4.119)$$

wird für einen Zustand, einen Sollzustand und eine Einstellzeit die Trajektorie des Roboterzustandes nach (4.118) gebildet. Dabei werden die Parameter der entsprechenden Polynome für das Überführen des Roboters aus dem aktuellen Zustand $\vec{q} = \vec{q}(0)$ in den Zielzustand $\vec{q}_r = \vec{q}(T_e)$ in der Einstellzeit T_e bestimmt.

4.3. Regelung

Mit Hilfe des Modells zur Bestimmung der Beschleunigungsfähigkeit kann nach Gleichung (4.107) für einen Zustand und eine geforderte Sollbeschleunigung die maximale Skalierung Λ bestimmt werden, für welche diese Beschleunigung realisiert werden kann. Ein Bewegungsprofil $\vec{q}(t) = \mathcal{Q}(\vec{q}, \vec{q}_r, T_e)$ ist dann valide, wenn $\forall t \in [0, T_e] : \Lambda(\vec{q}(t), \dot{\vec{q}}(t)) \geq 1$, wenn also an keinem Zeitpunkt die geforderte Beschleunigung verkleinert werden muss. Da sich ein Bewegungsprofil über (4.119) aus aktuellem und gefordertem Zustand sowie Einstellzeit ergibt, wird nach der kleinsten Einstellzeit T_e gesucht, für welche

$$\Lambda_{\min} : \mathfrak{Q} \times \mathfrak{Q} \times \mathbb{R} \rightarrow \mathbb{R}$$

$$(\vec{q}, \vec{q}_r, T_e) \mapsto \Lambda_{\min}(\vec{q}, \vec{q}_r, T_e) = \min_{0 \leq t \leq T_e} \left(\Lambda \left(\mathcal{Q}(\vec{q}, \vec{q}_r, T_e)(t), \dot{\mathcal{Q}}(\vec{q}, \vec{q}_r, T_e)(t) \right) \right) \quad (4.120)$$

Validität signalisiert – für die praktische Anwendung wird diese Überprüfung nur an einzelnen Stützstellen vorgenommen. Bei Vorgabe eines Startwertes für $T_e > 0$ kann mit Hilfe einer binären Suche eine beliebige Näherung an das kleinste valide T_e vorgenommen werden, siehe Algorithmus 9.

Algorithmus 9: Finde kleinste Einstellzeit

```

Eingabe : Zustandsübergang  $(\vec{q}, \vec{q}_r)$ , Startzeit  $T$ , Iterationszahl  $I$ 
1 Beginn
2   solange  $\Lambda_{\min}(\vec{q}, \vec{q}_r, T) < 1$  tue
3      $T \leftarrow 2T$ 
4    $a \leftarrow 0, b \leftarrow T, i \leftarrow 1$ 
5   solange  $i < I$  tue
6      $m \leftarrow \frac{1}{2}(a + b)$ 
7     wenn  $\Lambda_{\min}(\vec{q}, \vec{q}_r, m) > 1$  dann
8        $b \leftarrow m$ 
9     sonst
10       $a \leftarrow m$ 
11     $i \leftarrow i + 1$ 
  Ergebnis :  $b$ 

```

Die aktuelle Sollbeschleunigung des Roboters ergibt sich aus (4.119) zu

$$\ddot{\vec{q}}_r = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \dot{\mathcal{Q}}(\vec{q}, \vec{q}_r, T_e)(0), \quad (4.121)$$

bei der Suche nach der kleinsten Einstellzeit wird diese sowie die zugehörige Antriebskraftkonfiguration \vec{d} implizit bestimmt.

4. Regelung eines omnidirektionalen Antriebssystems

Ansatz der stückweise konstanten Beschleunigung

Der Ansatz der stückweise konstanten Beschleunigung wurde bereits in [Lan05] für den Spieler Lukas [Jah09] realisiert. Diesem Ansatz liegt die Lösung des Problems der zeitoptimalen Steuerung für ein lineares System mit zweidimensionalem Zustand und konstanter Begrenzung der Steuergröße zugrunde. Diese Annahme ist hier nicht erfüllt, dennoch wurde in [Lan05] praktische Eignung des Ansatzes für eine suboptimale Lageregelung nachgewiesen.

Analog zu (4.113) wird der Beschleunigungszeitverlauf $\ddot{\rho}(t)$ einer Koordinate ρ definiert. Während (4.113) einen linearen Beschleunigungsverlauf modelliert, werden nun zwei Phasen konstanter Beschleunigung gefordert:

$$\ddot{\rho}(t) = \begin{cases} \ddot{\rho}_0 & , t < T \\ \ddot{\rho}_e & , t \geq T. \end{cases} \quad (4.122)$$

Integration von (4.122) liefert das Geschwindigkeitsprofil

$$\dot{\rho}(t) = \int_0^t \ddot{\rho}(\tau) d\tau + \dot{\rho}_0 = \begin{cases} \ddot{\rho}_0 t + \dot{\rho}_0 & , t < T \\ (t - T)\ddot{\rho}_e + T\ddot{\rho}_0 + \dot{\rho}_0 & , t \geq T \end{cases} \quad (4.123)$$

und die Lage

$$\rho(t) = \int_0^t \dot{\rho}(\tau) d\tau + \rho_0 = \begin{cases} \frac{1}{2}t^2\ddot{\rho}_0 + t\dot{\rho}_0 + \rho_0 & , t < T \\ \left(\frac{1}{2}T^2 - tT\right)(\ddot{\rho}_e - \ddot{\rho}_0) + \frac{1}{2}t^2\ddot{\rho}_e + t\dot{\rho}_0 + \rho_0 & , t \geq T \end{cases} \quad (4.124)$$

Über die Randbedingungen $\dot{\rho}_e = \dot{\rho}(T_e)$ und $\rho_e = \rho(T_e)$ ergeben sich die Einstellzeit¹³ T_e

$$(T_e)_{1,2} = \frac{\dot{\rho}_e\ddot{\rho}_0 - \dot{\rho}_0\ddot{\rho}_e}{\ddot{\rho}_0\ddot{\rho}_e} \pm \frac{\sqrt{(2(\rho_e - \rho_0)\ddot{\rho}_0 + \dot{\rho}_0^2)\ddot{\rho}_e^2 + (2(\rho_0 - \rho_e)\ddot{\rho}_0^2 - (\ddot{\rho}_e^2 + \dot{\rho}_0^2)\ddot{\rho}_0)\ddot{\rho}_e + \dot{\rho}_e^2\ddot{\rho}_0^2}}{\ddot{\rho}_0\ddot{\rho}_e} \quad (4.125)$$

und die Umschaltzeit T

$$T = \frac{T_e\ddot{\rho}_e + \dot{\rho}_0 - \dot{\rho}_e}{\ddot{\rho}_e - \ddot{\rho}_0}. \quad (4.126)$$

¹³Sind $(T_e)_{1,2}$ beide positiv, dann wird die kleinere der beiden als Einstellzeit herangezogen. Ist nur eine der beiden positiv, so wird diese als Einstellzeit genutzt. Im Falle einer komplexen Lösung oder zweier negativer Werte, ist der Ansatz nicht anwendbar.

Zur Bestimmung der Umschaltzeiten werden Abschätzungen der möglichen Beschleunigungen benötigt. Die maximale Startbeschleunigung in Richtung der Solllage ergibt sich über das Beschleunigungsmodell¹⁴ zu

$$\ddot{\mathbf{q}}_0 = \Lambda \left(\ddot{\mathbf{q}}_0, \left(\dot{\mathbf{q}}_0^T, \frac{\mathbf{q}_r^T - \mathbf{q}_0^T}{\Delta T^2} \right)^T \right) \frac{\mathbf{q}_r^T - \mathbf{q}_0^T}{\Delta T^2} \quad (4.127)$$

und die maximale Bremsbeschleunigung

$$\ddot{\mathbf{q}}_e = k\Lambda \left(\ddot{\mathbf{q}}_r, \left(\dot{\mathbf{q}}_r^T, \frac{\mathbf{q}_0^T - \mathbf{q}_r^T}{\Delta T^2} \right)^T \right) \frac{\mathbf{q}_0^T - \mathbf{q}_r^T}{\Delta T^2}, \quad (4.128)$$

wobei der experimentell zu bestimmende Faktor¹⁵ $0 < k < 1$ eingeführt wird, um Überschwängen zu verhindern, welches durch die Annahme konstanter Beschleunigungsfähigkeit an diesem nichtlinearen System verursacht werden würde. Durch Auswertung der Gleichungen (4.126) und (4.125) werden die Umschaltzeiten T_x , T_y und T_Ψ für die einzelnen Komponenten der Bewegung aus den Beschleunigungen $\ddot{\mathbf{q}}_0$ und $\ddot{\mathbf{q}}_e$ bestimmt. Hiermit ergeben sich die Zeitverläufe $\rho_x(t)$, $\rho_y(t)$ und $\rho_\Psi(t)$ nach Gleichung (4.124). Bei direkter Verwendung der sich somit nach Gleichung (4.122) ergebenden Beschleunigung sind während des Bremsvorgangs jedoch hochfrequente Oszillationen der Beschleunigung und damit verbundene mechanische und elektrische Belastungen des Antriebssystems zu erwarten. Deshalb werden die erzeugten Zeitverläufe statt dessen indirekt verwendet, indem die Geschwindigkeit

$$\dot{\mathbf{q}}_r = \left(\dot{\rho}_x(T_x) \quad \dot{\rho}_y(T_y) \quad \dot{\rho}_\Psi(T_\Psi) \right)^T \quad (4.129)$$

als Führungsgröße eines unterlagerten Geschwindigkeitsreglers nach 4.3.1 herangezogen wird. Dieser glättet den resultierenden Beschleunigungsverlauf, siehe Abbildung 4.15(e).

Die Antworten auf einen Führungssprung der beiden implementierten Lage-regler finden sich in Abbildung 4.15. Die Spur des Roboterlage bei einem Führungssprung mit verschiedenen Startgeschwindigkeiten ist in Abbildung

¹⁴ In den Gleichungen (4.127) und (4.128) werden die maximalen Beschleunigungen entlang bzw. entgegen des Lagefehlers bestimmt. Deshalb ist für diesen Anwendungsfall die Begrenzung der Skalierungsvariable nicht Teil der Optimierungsaufgabe. Der Term $\Delta T \neq 0$ dient lediglich der Anpassung der Einheiten.

¹⁵ Zur Bestimmung des Verstärkungsfaktors wird eine für den Anwendungsbereich repräsentative Menge von Führungssprüngen untersucht, siehe Abbildung 4.14. Es wird dabei ein Überschwängen der Position um ε toleriert. Solange bei mindestens einem Sprung nicht toleriertes Überschwängen auftritt, ist der Verstärkungsfaktor invalide. Durch eine binäre Suche ermittelte Verstärkungsfaktoren für das Modell nach Tabelle 4.4 finden sich in Tabelle 4.5.

4. Regelung eines omnidirektionalen Antriebssystems

4.16 dargestellt. Die Abbildungen 4.15 und 4.16 zeigen, dass das Ziel der Lage-
regelung mit beiden Varianten erfüllt wird. Die realisierten Regler überführen
den Roboter unter Berücksichtigung der Beschleunigungsgrenzen ohne Über-
schwingen aus einer Ausgangslage in in eine Ziellage. Dabei hat der Regler auf
Grundlage der stückweise konstanten Beschleunigung leichte Vorteile in Bezug
auf die Ausregelzeit.

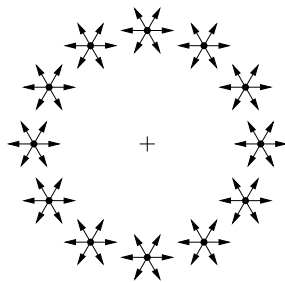
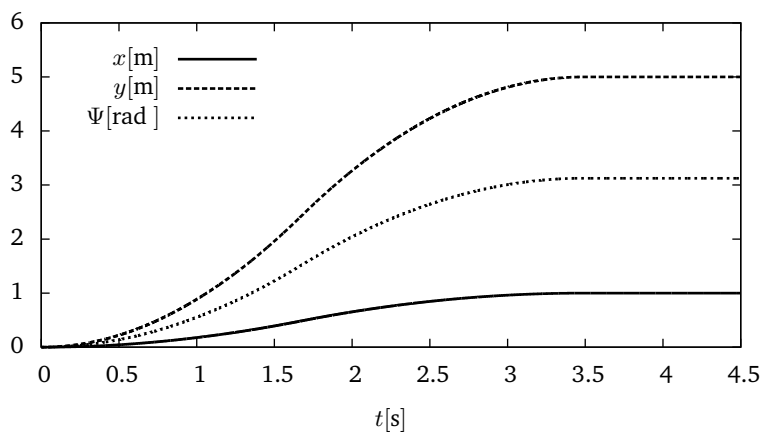


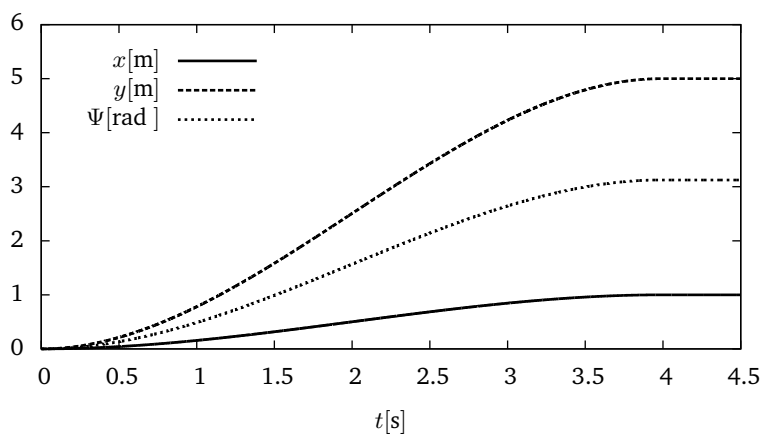
Abbildung 4.14.: Testraster zur Bestimmung der Verstärkung der Bremsbeschleunigung: Auf Suchkreisen der Radien 20cm, 40cm,..., 260cm werden je 12 Positionen mit je 6 Zielausrichtungen angefahren.

$\Delta T[\text{ms}]$	$\varepsilon[\text{mm}]$	k	$\bar{v}[\text{ms}^{-1}]$
10	50	0,82	0,77
10	25	0,77	0,75
10	10	0,71	0,72
10	5	0,66	0,69
10	2,5	0,52	0,63
1	50	0,88	0,77
1	25	0,84	0,75
1	10	0,80	0,74
1	5	0,77	0,73
1	2,5	0,76	0,73
10	<1	—*	0,63

Tabelle 4.5.: Bestimmung des Verstärkungsfaktors der Bremsbeschleunigung für vorgegebene ε -Toleranz durch binäre Suche, mittlere planare Geschwindigkeit über die Menge der Testfälle.
* Mittlere Geschwindigkeit bei linearer Beschleunigung.

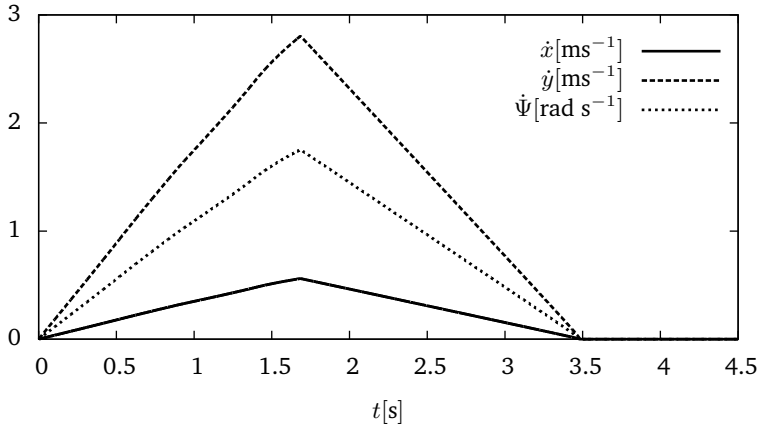


(a) Zeitverlauf der Lage bei stückweise konstanter Beschleunigung

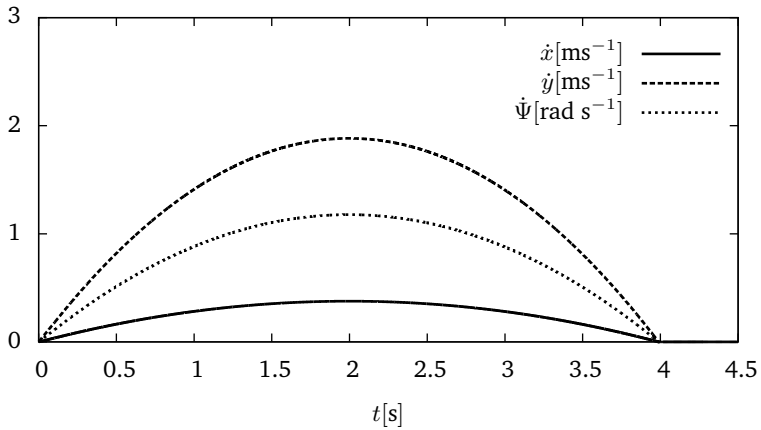


(b) Zeitverlauf der Lage bei linearer Beschleunigung

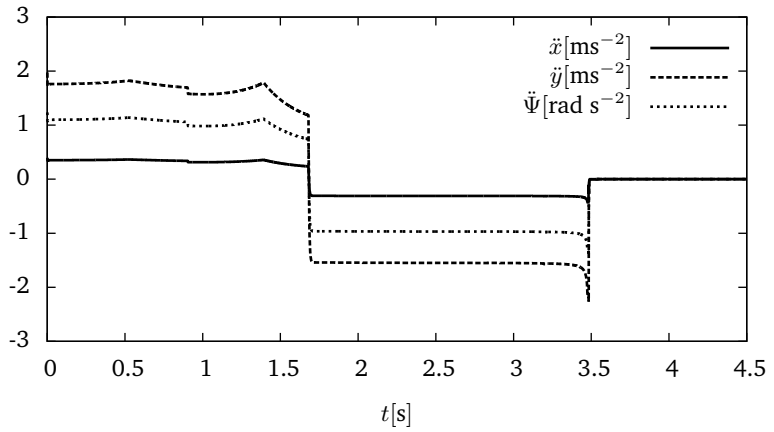
4. Regelung eines omnidirektionalen Antriebssystems



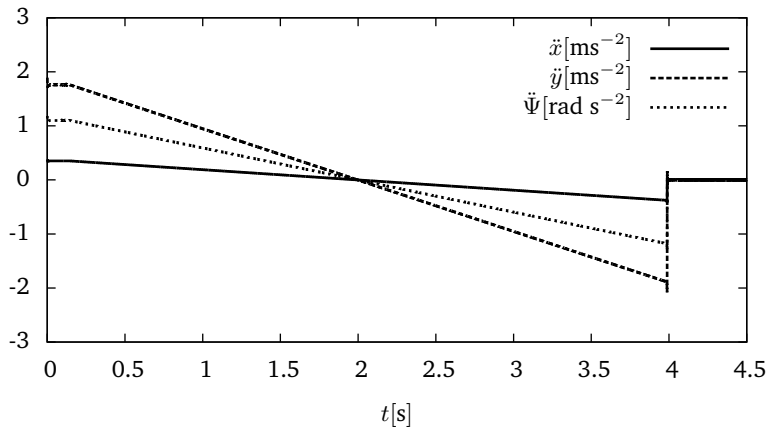
(c) Zeitverlauf der Geschwindigkeit bei stückweise konstanter Beschleunigung



(d) Zeitverlauf der Geschwindigkeit bei linearer Beschleunigung



(e) Zeitverlauf der Beschleunigung bei stückweise konstanter Beschleunigung



(f) Zeitverlauf der Beschleunigung bei linearer Beschleunigung

Abbildung 4.15.: Sprung der Führungsgröße am Lageregler.

4. Regelung eines omnidirektionalen Antriebssystems

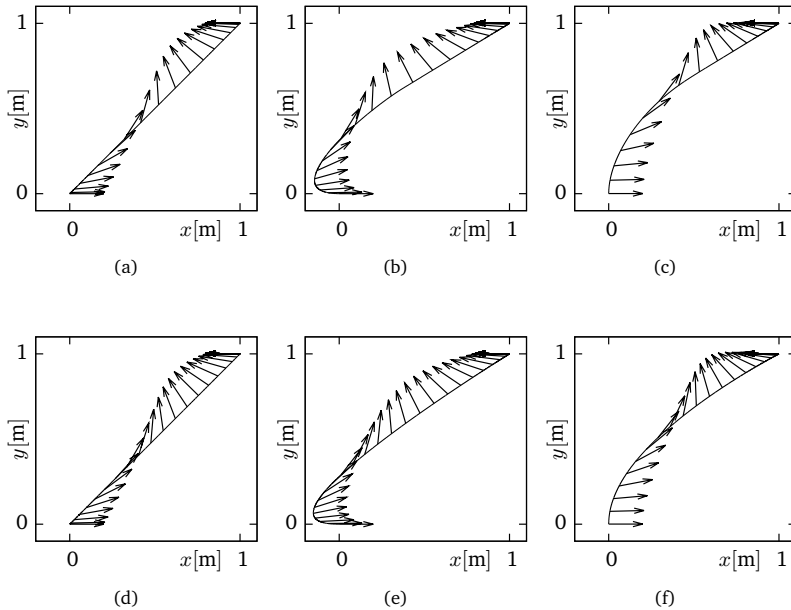


Abbildung 4.16.: Spur des lagegeregelten Roboters bei Führungssprung. Lage-regler mit stückweise konstanter Beschleunigung (oben) sowie linearem Beschleunigungsverlauf (unten). Roboter zu Beginn im Stillstand (links), mit Geschwindigkeitskomponente in Richtung \vec{E}_x (Mitte) und mit Geschwindigkeitskomponente in Richtung \vec{E}_y (rechts). Der Pfeil zeigt in Richtung \vec{E}_x .

5. Pfadplanung

Für das Überführen eines Roboters in eine Ziellage wurden in Kapitel 4 Lageregler für ein omnidirektionales Antriebssystem dargelegt. Diese nichtlinearen Regler basieren auf dem Modell zur Beschreibung der Beschleunigungsfähigkeit unter Berücksichtigung der elektrischen und mechanischen Nebenbedingungen aus Abschnitt 4.2. Beim Einsatz des Roboters im autonomen Fußballspiel existieren zusätzlich geometrische Nebenbedingungen, diese sind durch eine wichtige Forderung formuliert: Zum Ausfall eines Spielers durch Beschädigung oder Sanktionen des Schiedsrichters führende Kollisionen müssen vermieden werden. Die Roboter können als zylindrische Objekte betrachtet werden und verfügen über omnidirektionale Antriebssysteme. Um Kollisionen mit den Mitspielern und den Spielfeldgrenzen zu vermeiden, ist die Validität der Trajektorien aller Punkte eines Roboters sicherzustellen, siehe Abbildung 5.1(a).

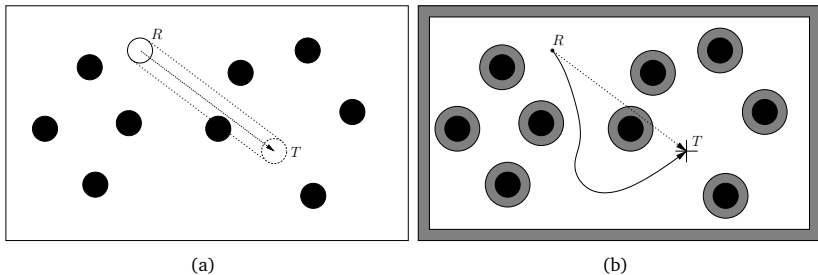


Abbildung 5.1.: Bewegung eines Roboters auf einem Spielfeld mit Hindernissen

Die wichtigste Vereinfachung dieses Problems ergibt sich durch die omnidirektionalen Antriebssysteme. Sie ermöglichen es, die Planung als zweidimensionales Problem zu formulieren, da Translationen des Roboters unabhängig von der Orientierung ausgeführt werden können.

Die zweite Vereinfachung ergibt sich aus der zylindrischen Form der zu bewegendenden Roboter. Durch die Erweiterung aller Hindernisgrenzen um den Radius des betrachteten Roboters (Abbildung 5.1(b)) muss lediglich die Validität der Trajektorie des Mittelpunktes des Roboters sichergestellt werden. Neben der

5. Pfadplanung

Forderung nach Validität können an die zu bestimmende Trajektorie Anforderungen wie *kürzester Weg*, *kurzer Weg mit großem Hindernisabstand*, *Zeitoptimalität* oder *Verbrauchsoptimalität* gestellt werden.

Die dritte Vereinfachung ergibt sich bei Verwendung der in Abschnitt 4.3.2 entworfenen Lageregelung. Sie erlaubt eine diskretisierte Beschreibung der gesuchten Trajektorie als Sequenz von Wegpunkten.

Erfolgreiche Varianten zur Erzeugung valider Trajektorien im Kontext des RoboCup stehen mit den Gradientenverfahren [Top99] bereit. Problematisch beim Einsatz von Gradientenverfahren ist dabei die Existenz lokaler Minima in den zugrunde liegenden Potentialfeldern. Bei Unterstützung durch Graphensuchverfahren ergeben sich jedoch praxistaugliche Lösungen [Top99]. Robuste Alternativen bestehen in der Auswertung von Regelwerken. Zur Formulierung dieser Regelwerke bieten sich beispielsweise Verbunde endlicher Automaten [Bro90] oder Methoden der unscharfen Logik [DM02] an. Ein Problem bei der Formulierung der Regelwerke ist es, Vollständigkeit der resultierenden Lösungen sicherzustellen.

Eine weitere, im RoboCup verbreitete, Lösung des Planungsproblems liegt bei den „Rapidly-exploring Random Trees“ [LK99, BV02, LaV06, Car06]. Diese stichprobenbasierten Planer ermitteln einen Pfad als Sequenz von Wegpunkten. Sie weisen eine hohe Verarbeitungsgeschwindigkeit auf, liefern dabei jedoch suboptimale Pfade.

Durch die genannten Vereinfachungen reduziert sich die Komplexität des Planungsproblems jedoch so weit, dass mit Hilfe kombinatorischer Methoden eine Optimallösung auch unter Echtzeitanforderungen gefunden werden kann. Die kombinatorische Planung basiert auf einem Navigationsgraphen. Dessen Knoten entsprechen validen Roboterpositionen, seine Kanten stellen valide Verbindungen zwischen diesen Knoten dar. Durch die Konstruktion eines Navigationsgraphen aus den Informationen über die Umgebung wird das Planungsproblem auf eine Graphensuche reduziert. Für die Graphensuche wird der A*-Algorithmus verwendet. In den Abschnitten 5.2 und 5.3 werden zwei Möglichkeiten zur Konstruktion eines Navigationsgraphen dargelegt. Für die Realisierung des Sichtbarkeitsgraphen in Abschnitt 5.2 werden die Hindernisse durch Polygone approximiert. Im resultierenden Graphen wird nach dem kürzesten Weg gesucht. Eine Variante eines Freiflächenplaners durch adaptive Rasterung wird in Abschnitt 5.3 gezeigt. Hierbei wird das Spielfeld in freie Zellen zerlegt, diese Zellen und deren passierbare Verknüpfungen sind in einem Quadtree organisiert und bilden den Navigationsgraphen.

5.1. Graphensuche

5.1.1. Definitionen

Für den Ansatz der kombinatorischen Planung kann das Planungsproblem als Suchproblem auf Graphen formuliert werden. Dadurch lassen sich eine Vielzahl unterschiedlicher Probleme mit einem gemeinsamen Formalismus lösen. Die Graphensuche zum Lösen des Pfadplanungsproblems basiert auf den Definitionen 8-14 aus [HNR68]:

Definition 8 Ein Graph G sei definiert über eine Menge $\{n_i\}$ von Knoten und über eine Menge $\{e_{ij}\}$ gerichteter Verbindungen zwischen Knoten.

Definition 9 Wenn $e_{pq} \in \{e_{ij}\}$, gibt es eine Verbindung von n_p zu n_q . n_q ist dann der Nachfolger von n_p .

Definition 10 Die Kosten einer Transition von Knoten n_i zu Knoten n_j über die Verbindung e_{ij} seien c_{ij} .

Eine übliche Repräsentation von Graphen stellt die Adjazenzmatrix dar, in welcher die Nachbarschaftsbeziehungen bzw. Transitionskosten hinterlegt sind, siehe Abbildung 5.2.

Für die im Folgenden dargestellte Graphensuche mittels A*-Algorithmus ist es jedoch nicht notwendig, den kompletten Graphen explizit durch eine Adjazenzmatrix zu repräsentieren, statt dessen kann auf eine implizite Darstellung nach [HNR68] zurückgegriffen werden, welche durch einen Startknoten s und den Nachfolgeroperator Γ definiert ist. Der Nachfolgeroperator Γ liefert für einen beliebigen Knoten n_i eine Menge von Paaren bestehend aus erreichbaren Knoten n_j und den zugehörigen Transitionskosten c_{ij} :

$$\begin{aligned} \Gamma : \{n_i\} &\rightarrow \{\{(n_j, c_{ij})\}\} \\ n_i &\mapsto \Gamma(n_i) = \{(n_j, c_{ij})\}. \end{aligned} \tag{5.1}$$

Die Anwendung des Nachfolgeroperators Γ auf einen Knoten n_i wird als Expansion des Knotens n_i bezeichnet.

Definition 11 Ist der Subgraph G_n durch einen Startknoten $n \in \{n_i\}$ und den Nachfolgeroperator Γ implizit definiert, dann heißen alle Knoten aus G_n erreichbar von n .

Definition 12 Ein Pfad von n_1 bis n_k ist eine geordnete Menge von Knoten, bei der jeder Knoten n_{i+1} Nachfolger von Knoten n_i ist.

5. Pfadplanung

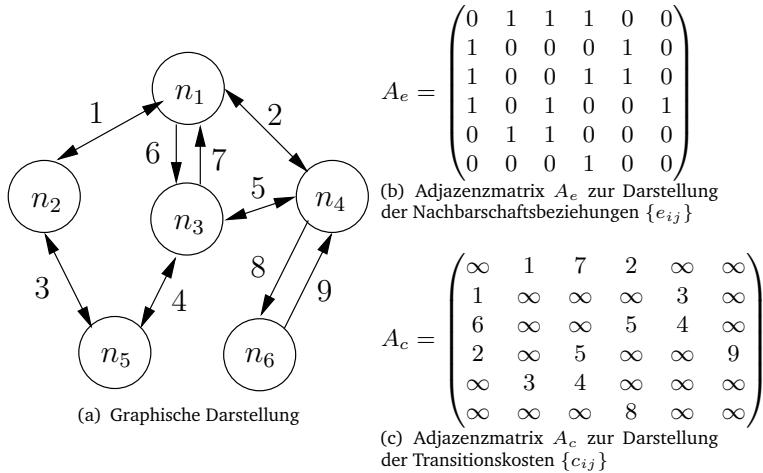


Abbildung 5.2.: Beispiel eines gewichteten und gerichteten zyklischen Graphen.

Definition 13 Die Kosten g eines jeden Pfades sind die Summe aller Transitionskosten $c_{i,i+1}$ zwischen den einzelnen Knoten des Pfades.

Definition 14 Die Kosten des Pfades von einem festen Startknoten $s \in \{n_i\}$ zu einem Knoten $n \in \{n_i\}$ heißen $g(n)$.

5.1.2. A*-Algorithmus

Der A*-Algorithmus [HNR68] ist ein informiertes bzw. heuristisches Graphen-suchverfahren, es bezieht Wissen aus dem Problembereich in die Suche ein.

Der Algorithmus ist vollständig, optimal und optimal effizient: Falls im Graphen G ein Pfad vom Startknoten n_u zum Zielknoten n_v existiert, wird er durch den Algorithmus gefunden. Jeder gefundene Pfad hat minimale Kosten, das heißt, dass in G kein anderer Pfad von n_u nach n_v mit geringeren Kosten existiert. Für die Bestimmung des Pfades wird dabei eine minimale Anzahl an Knoten expandiert, es kann laut [HNR68] keinen Algorithmus geben, der auf dem selben Graphen unter Nutzung der selben Heuristik einen optimalen Pfad bestimmt und dabei weniger Knoten expandieren muss.

Der A*-Algorithmus (Algorithmus 10 nach [HNR68]) findet in einem Graphen G_s den optimalen Pfad von einem Startknoten s zu einem Zielknoten $t \in T$,

wobei T die Menge aller erlaubten Zielknoten ist. Hierfür benutzt er eine Funktion $f(n)$ zur Auswahl der zu expandierenden Knoten.

Algorithmus 10: A*

```

Eingabe : Startknoten  $s$ , Menge möglicher Zielknoten  $T$ 
1 Beginn
2   Markiere  $s$  als „offen“ und berechne  $f(s)$ .
3   Wähle den offenen Knoten  $n$  dessen  $f$  minimal ist.
4   wenn  $n \in T$  dann
5     Markiere  $n$  als geschlossen
     Ergebnis : Durch Traversal von  $n$  über alle Vorgänger bis  $s$  erhaltener
       Pfad
6   sonst
7     Markiere  $n$  als geschlossen und wende den Nachfolgeroperator  $\Gamma$  auf  $n$ 
       an
8     Berechne  $f(n_i)$  für alle Nachfolger  $n_i$  von  $n$ .
9     Markiere alle nicht bereits geschlossenen Nachfolger  $n_i$  als offen.
10    Bereits geschlossene Nachfolger  $n_i$  werden wieder als offen markiert,
       wenn ihr aktuelles  $f(n_i)$  kleiner ist, als bei dem Schritt, in dem sie als
       geschlossen markiert wurden
11    Speichere  $n$  für alle offenen Nachfolger  $n_i$  von  $n$  als Vorgänger.
12    wenn Es gibt offene Knoten dann
13      | Gehe weiter zu Zeile 3.
14    sonst
      | Ergebnis : Es existiert Kein Pfad

```

Als Auswahlfunktion $f(n)$ wird die Summe der tatsächlichen Kosten $g(n)$ des Pfades vom Startknoten s zum Knoten n und der erwarteten Kosten $h(n, t)$ von n zum Zielknoten t benutzt:

$$f : \{n_i\} \rightarrow \mathbb{R} \quad (5.2)$$

$$n \mapsto f(n) = g(n) + h(n, t)$$

Abbildung 5.3 auf Seite 136 zeigt die Funktionsweise des Algorithmus an einem Beispiel.

Notwendige Voraussetzung für die Einsetzbarkeit des A*-Algorithmus auf einem Graphen G ist, dass alle Kantengewichte bzw. Transitionskosten c_{ij} von G positiv sind. Weiterhin wird eine Heuristik benötigt, welche das Wissen aus der Problemdomäne für die Graphensuche nutzbar macht. Sie beschleunigt den Algorithmus gegenüber uninformierten Suchverfahren wie Breiten-, uniforme Kosten- und Tiefensuche und hat entscheidenden Einfluss auf die Effizienz der Pfadberechnung. Die Heuristik liefert eine Schätzung der zu erwartenden Kosten von einem beliebigen Knoten n_i zum Zielknoten t . In [RN95] wird zwischen

5. Pfadplanung

A	g	h	f	P
0	5	5	—	

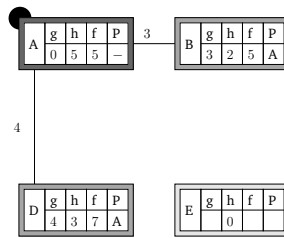
E	g	h	f	P
	0			

unbesucht

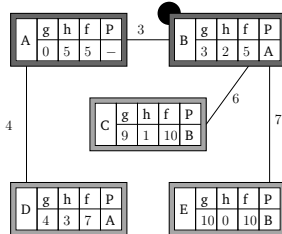
offen

geschlossen

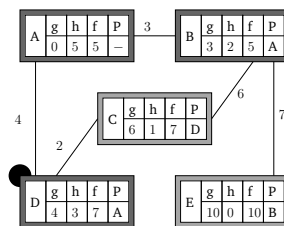
(a) Gesucht ist der günstigste Pfad von *A* nach *E* in einem gewichteten Graphen.



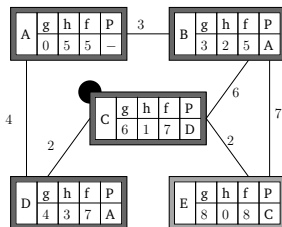
(b) Der Startknoten *A* ist aktiv und wird expandiert. Dabei werden die Werte für *g*, *h*, *f* und *V* der erreichbaren Nachbarknoten *B* und *D* bestimmt. Der offene Knoten mit kleinstem Wert *f* ist nun *B*, dieser wird im nächsten Schritt aktiv.



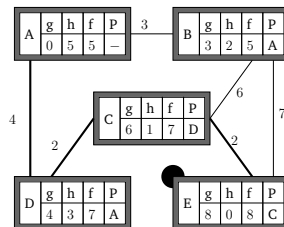
(c) Knoten *B* ist aktiv, wird expandiert und geschlossen. Hierbei ergibt sich *B* als Vorgänger von *C* und *E*. Knoten *D* ist nun der offene Knoten minimalen Wertes von *f* – er wird im nächsten Schritt aktiv.



(d) Es ergibt sich ein günstigerer Pfad nach *C* über *D*, Knoten *C* wird aktualisiert. *C* ist der offene Knoten mit minimalem *f*.



(e) Bei der Expansion von *C* ergibt sich kein günstigerer Weg nach *B*, *B* bleibt unverändert. Knoten *E* wird aktualisiert.



(f) Der aktive Knoten ist der Zielknoten, der gesuchte Pfad ergibt sich durch Rückverfolgung der Vorgängerknoten.

Abbildung 5.3.: Beispiel A*

5.2. Aufbau des Navigationsgraphen als Sichtbarkeitsgraph

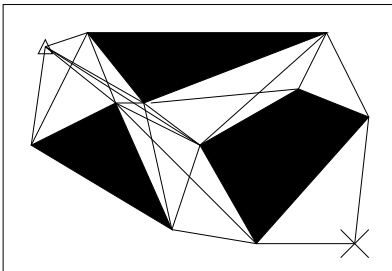
zulässigen und monotonen Heuristiken unterschieden. Zulässig sind alle Heuristiken, die die Kosten zum Zielknoten nie überschätzen. Monotone Heuristiken genügen zusätzlich der Dreiecksungleichung, so dass die Auswahlfunktion f entlang eines Pfades niemals abnimmt. Die für geometrische Planungsaufgaben häufig genutzte Heuristik der euklidischen Distanz ist monoton.

Als Navigationsgraph wird im Folgenden ein als Graph der beschriebenen Eigenschaften formuliertes Planungsproblem bezeichnet.

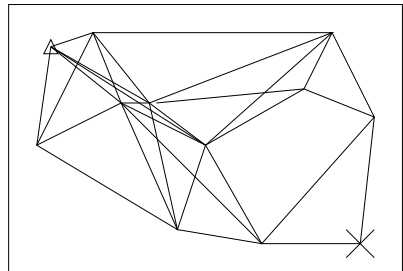
5.2. Aufbau des Navigationsgraphen als Sichtbarkeitsgraph

Beim Ansatz des Sichtbarkeitsgraphen[Nil69] wird der kürzest mögliche Weg entlang der Schattenlinien um die Hindernisse herum gesucht. Der Ansatz ist weit verbreitet für Anwendungen, bei denen kleine Abstände zu den Hindernissen toleriert werden können [Wei98, Top99, Eic07, dCvO08].

In [Eic07] wird gezeigt, wie ein Sichtbarkeitsgraph für kreisförmige Hindernisse exakt aufgebaut werden kann. Die Approximation der Hindernisse durch konvexe Polygone (siehe Abbildung 5.4) erlaubt jedoch eine geschwindigkeitsabhängige Polygone deformation um variable Sicherheitsabstände zu modellieren. Dem gegenüber steht der Nachteil, dass die Planungsaufgabe durch die Approximation nicht mehr exakt optimal gelöst werden kann.



(a) Start- und Ziel- sowie die Eckpunkte aller Hindernisse bilden die Knoten des Navigationsgraphen. Strecken zwischen diesen Knoten sind Kanten, wenn sie kein Hindernis schneiden.



(b) Der kürzeste Weg vom Startpunkt zum Zielpunkt ist in diesem Graphen enthalten und kann durch eine Graphensuche bestimmt werden.

Abbildung 5.4.: Konstruktion des Sichtbarkeitsgraphen in der Spielfeldfläche

Für den Roboter der Position R soll der kürzeste Pfad zur Zielposition T geplant werden. Dabei sind die konvexen Hindernisspolygone $\mathcal{P}_0, \dots, \mathcal{P}_{m-1}$ zu berücksichtigen.

5. Pfadplanung

sichtigen, keine Teilstrecke des Pfades darf ein Hindernis schneiden und kein Wegpunkt darf innerhalb eines Hindernisses liegen.

Definition 15 Das Polygon \mathcal{P} der Eckpunkte (P_0, \dots, P_{k-1}) , $P_i \in \mathbb{R}^2$ sei die Menge der von den Kanten $\mathcal{K}_0, \dots, \mathcal{K}_{k-1}$ linksseitig umschlossenen Punkte. Dabei sei die Kante $\mathcal{K}_j := (P_j, P_{(j+1) \bmod k})$ die durch zwei aufeinanderfolgende Eckpunkte beschriebene Strecke.

Für den Aufbau des Sichtbarkeitsgraphen werden die Hindernispolygone \mathcal{P}_i als offene Mengen betrachtet. Aus dieser Problembeschreibung ergibt sich die Menge der Navigationspunkte

$$N = \{n_i\} := \{R, T, P_{0,0}, \dots, P_{m-1,k_{m-1}-1}\} \setminus (\mathcal{P}_0 \cup \dots \cup \mathcal{P}_{m-1}). \quad (5.3)$$

Sind Start- oder Zielpunkt nicht in der Menge der Navigationspunkte enthalten, ist die Planungsaufgabe nicht lösbar. Andernfalls bildet die Menge der Navigationspunkte die Knoten des Navigationsgraphen G . Die Kanten $\{e_{i,j}\}$ des Graphen sind die geometrisch validen Verbindungen $S := (n_p, n_q)$ zweier Navigationspunkte,

$$e_{p,q} \in \{e_{i,j}\}, \text{ wenn } (n_p, n_q) \text{ schneidet kein } \mathcal{P} \in \{\mathcal{P}_0, \dots, \mathcal{P}_{m-1}\}. \quad (5.4)$$

Die Kosten $c_{p,q}$ einer Kante $e_{p,q}$ und die Heuristik $h(p, t)$ ergeben sich als aus dem Abstand der entsprechenden Navigationspunkte

$$c_{p,q}^1 := \|\overline{n_p n_q}\|_2, \quad (5.5)$$

$$h(p, t) := \|\overline{n_p n_t}\|_2. \quad (5.6)$$

In der vorliegenden Implementation hat der Aufbau des vollständigen Sichtbarkeitsgraph eine asymptotische Laufzeit $O(n^3)$, wobei $n = |N|$. Dieser Wert resultiert aus dem naiven² Ansatz der Implementation, ist aber wegen der kleinen Problemgröße und durch den impliziten Aufbau nicht relevant für die Anwendung. Der Vorteil des impliziten Aufbaus liegt darin, dass der Sichtbarkeitsgraph nicht vollständig, sondern nur an den benötigten Stellen bestimmt wird. In der praktischen Anwendung wurden Laufzeiten³ im Bereich 0,5 ms

¹Die Kosten aller Nachfolger des Startknotens können um eine Winkelstrafe erhöht werden. Diese bestraft Veränderungen der Orientierung bezüglich der tatsächlichen oder der zuletzt geplanten Bewegungsrichtung. Eine solche Winkelstrafe kann als Hysterese des Planers betrachtet werden. Sie unterdrückt Oszillationen zwischen Lösungen niedriger Differenz der Gesamtkosten.

²In [GM91] findet sich ein Algorithmus zur Bestimmung des Sichtbarkeitsgraphen für Hindernispolygone mit der asymptotischen Laufzeit von $O(n \log n + k)$, wobei $n = |N|$, $k = |\{e_{i,j}\}|$.

³Intel®Core™2 Duo CPU E6850, 3GHz, 5 Eckpunkte pro Polygon, 10 Hindernisse

5.2. Aufbau des Navigationsgraphen als Sichtbarkeitsgraph

pro Pfad erzielt, weswegen Verbesserungen an dieser Stelle momentan nicht erforderlich erscheinen.

5.2.1. Approximation der Hindernisse durch Polygone

Die Approximation der Hindernisse durch Polygone geschieht anhand der Roboterposition $\vec{r} = (r_x, r_y)$, der Hindernisposition $\vec{o} = (o_x, o_y)$, dem Mindestabstand a zwischen diesen Positionen sowie der Anzahl der Eckpunkte k . Bei der Approximation wird das Polygon am Abstandsvektor $\vec{d} := (\vec{r} - \vec{o})$ orientiert. Hierzu wird die Kante \mathcal{K}_{k-1} orthogonal zum Abstandsvektor ausgerichtet. Die Eckpunkte (P_0, \dots, P_{k-1}) zu

$$\vec{p}_i = a \sqrt{\tan^2\left(\frac{\pi}{k}\right) + 1} \begin{pmatrix} \cos\left(\pi \frac{1+2i}{k}\right) & -\sin\left(\pi \frac{1+2i}{k}\right) \\ \sin\left(\pi \frac{1+2i}{k}\right) & \cos\left(\pi \frac{1+2i}{k}\right) \end{pmatrix} \frac{(\vec{o} - \vec{r})}{|\vec{o} - \vec{r}|} + \vec{o}. \quad (5.7)$$

beschreiben das Hindernis als konvexes Polygon, siehe Abbildung 5.5(a). Die so ermittelten Polygone können durch eine heuristisch aus Abstand zum Hindernis und Objektgeschwindigkeiten ermittelte Verschiebung $\vec{\Delta}$ deformiert werden, siehe Abbildung 5.5(b) und 5.5(c).

5. Pfadplanung

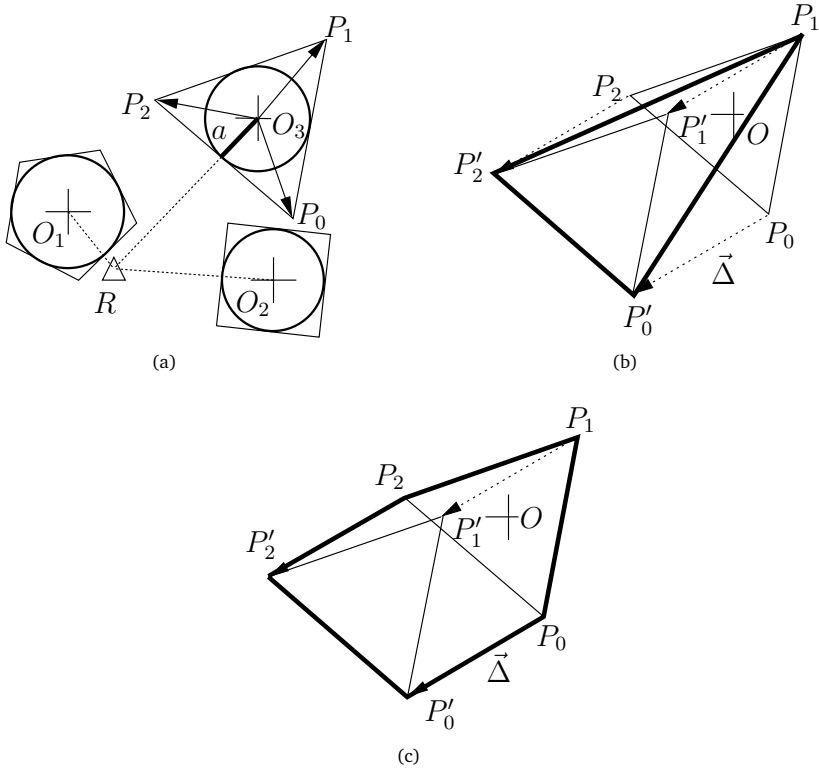


Abbildung 5.5.: Approximation von Hindernissen durch konvexe Polygone (a) und geschwindigkeitsabhängige Polygone deformation durch heuristisch bestimmte Verschiebung $\vec{\Delta}$ (b,c). Gleichbleibende Eckpunktzahl durch bedingte Verschiebung: Ist $(\vec{p}_i - \vec{o})\vec{\Delta} > 0$, dann $\vec{p}_i \leftarrow \vec{p}_i + \vec{\Delta}$ (a). Durch konvexe Hülle (b)

5.2.2. Punkt-im-Polygon-Test

Die Normalisierung eines Winkels φ auf das Intervall $[-\pi, \pi)$ geschehe anhand

$$\begin{aligned} \Phi : \mathbb{R} &\rightarrow \mathbb{R} \\ \varphi &\mapsto \Phi(\varphi) = ((\varphi + \pi) \bmod 2\pi) - \pi, \end{aligned} \quad (5.8)$$

wobei die Operation $a \bmod c$ mit $a, c \in \mathbb{R}$, entsprechend den Definitionen

Definition 16 Für $a, b, c \in \mathbb{R}, c \neq 0$ gilt, dass a und b kongruent mod c sind, wenn $a - b = kc$ mit $k \in \mathbb{Z}$.

Definition 17 Für $a, c \in \mathbb{R}, c \neq 0$ sei $a \bmod c$ die Zahl, die kongruent mod c ist und in $[0, c)$ liegt.

zu

$$a \bmod c = a - c \left\lfloor \frac{a}{c} \right\rfloor \quad (5.9)$$

dargestellt wird, vergleiche [Knu97]. Der normalisierte Winkel β_i zwischen zwei aufeinanderfolgenden Eckpunkten $P_i, P_{(i+1) \bmod k}$ bezüglich eines Probepunktes A ergibt sich zu

$$\beta_i = \Phi(\arg(\vec{p}_i - \vec{a}) - \arg(\vec{p}_{(i+1) \bmod k} - \vec{a})), \quad (5.10)$$

wobei die Orientierung eines Vektors $\vec{r} \in \mathbb{R}^2$ mit dem quadrantenbereinigten Arkustangens $\arg(\vec{r})$ nach Gleichung (3.7) auf Seite 57 bestimmt wird. Anhand $\sigma = \sum_{i=0}^k \lfloor \beta_i \rfloor$ lässt sich nach [BM01] eine Aussage über die Lage von A treffen:

$$A \text{ liegt } \begin{cases} \text{außerhalb } \mathcal{P} & , \sigma = 0, \\ \text{innerhalb } \mathcal{P} & , \sigma = 2\pi, \\ \text{auf einer Kante von } \mathcal{P} & , \sigma = \pi, \\ \text{auf einem Eckpunkt von } \mathcal{P} & , \text{sonst,} \end{cases} \quad (5.11)$$

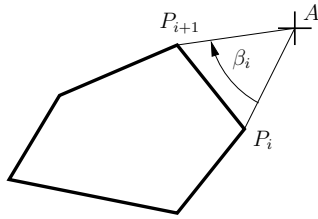
siehe Abbildung 5.6.

5.2.3. Strecke-schneidet-Polygon-Test

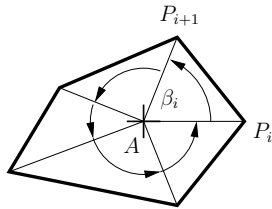
Für den Aufbau des Sichtbarkeitsgraphen ist die Überprüfung notwendig, ob eine Strecke $\mathcal{S} := (S_1, S_2)$, $S_1, S_2 \in \mathbb{R}^2$ das Polygon \mathcal{P} schneidet. Diese Über-

⁴ $\lfloor x \rfloor$ ist die Abrundungsfunktion, für die gilt: $\lfloor x \rfloor := \max_{k \in \mathbb{Z}, k \leq x} (k)$.

5. Pfadplanung



(a) A liegt außerhalb.



(b) A liegt innerhalb.

Abbildung 5.6.: Punkt-im-Polygon-Test

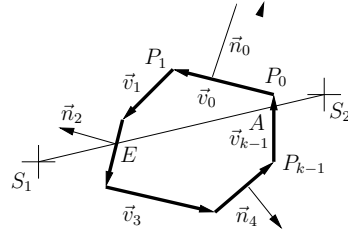


Abbildung 5.7.: Segment-Polygon-Schnitt

prüfung wird nach [Sun] durchgeführt. Hierzu werden \mathcal{S} und die Kanten \mathcal{K}_i des konvexen Polygons \mathcal{P} in die Geradengleichungen

$$\vec{s}(\lambda_s) = \vec{s}_1 + \lambda_s \vec{u}, \quad \vec{u} := \vec{s}_2 - \vec{s}_1 \quad (5.12)$$

und

$$\vec{k}_i(\lambda_i) = \vec{p}_i + \lambda_i \vec{v}_i, \quad \vec{v}_i := \vec{p}_{(i+1) \bmod k} - \vec{p}_i \quad (5.13)$$

überführt. Durch Rotation der Richtungsvektoren \vec{v}_i um $-\frac{1}{2}\pi$ ergeben sich die nach außen zeigenden Normalenvektoren \vec{n}_i der Kanten \mathcal{K}_i (siehe Abbildung 5.7):

$$\vec{n}_i := \begin{pmatrix} \cos\left(-\frac{\pi}{2}\right) & -\sin\left(-\frac{\pi}{2}\right) \\ \sin\left(-\frac{\pi}{2}\right) & \cos\left(-\frac{\pi}{2}\right) \end{pmatrix} \vec{v}_i = \begin{pmatrix} v_y \\ -v_x \end{pmatrix}_i \quad (5.14)$$

Hierdurch lässt sich der Geradenparameter $\lambda_{s,i}$ für den Schnittpunkt von \mathcal{S} und \mathcal{K}_i zu

$$\lambda_{s,i} = \frac{(\vec{p}_i - \vec{s}_1) \cdot \vec{n}_i}{\vec{u} \cdot \vec{n}_i} \quad (5.15)$$

5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition

bestimmen. Nach [Sun] ist anhand des Skalarproduktes $\vec{u}\vec{n}_i$ eine Unterscheidung nach eindringenden, verlassenden und parallelen Paarungen $(\mathcal{S}, \mathcal{K}_i)$ möglich:

$$\begin{aligned}\vec{u}\vec{n}_i < 0 &\implies \mathcal{S} \text{ dringt über } \mathcal{K}_i \text{ in } \mathcal{P} \text{ ein ,} \\ \vec{u}\vec{n}_i > 0 &\implies \mathcal{S} \text{ verlässt } \mathcal{P} \text{ über } \mathcal{K}_i, \\ \vec{u}\vec{n}_i = 0 &\implies \mathcal{S} \parallel \mathcal{K}_i.\end{aligned}\tag{5.16}$$

Damit ergeben sich nach [Sun] die Spurparameter λ_E, λ_A bezüglich \mathcal{S} für Eintritt bzw. Austritt:

$$\lambda_E = \max(0, \max_{0 \leq i < k \mid \vec{u}\vec{n}_i < 0} (\lambda_{s,i})),\tag{5.17}$$

$$\lambda_A = \min(1, \min_{0 \leq i < k \mid \vec{u}\vec{n}_i > 0} (\lambda_{s,i})).\tag{5.18}$$

Im Falle der Durchdringung von \mathcal{S} und \mathcal{P} stehen die Schnittpunkte anhand Gleichung (5.12) zur Verfügung. Die Überprüfung auf Durchdringung geschieht nach [Sun] zu

$$\begin{aligned}\exists i \mid (\vec{u}\vec{n}_i = 0) \wedge ((\vec{p}_i - \vec{s}_1) \vec{n}_i) \leq 0 &\implies \mathcal{S} \text{ schneidet } \mathcal{P} \text{ nicht,} \\ \lambda_E > \lambda_A &\implies \mathcal{S} \text{ schneidet } \mathcal{P} \text{ nicht.}\end{aligned}\tag{5.19}$$

5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition

Eine Alternative zum Aufbau des Navigationsgraphen als Sichtbarkeitsgraph ist die Bestimmung freier und konvexer Gebiete bzw. Zellen der Spielfeldfläche (Knoten) sowie deren Nachbarschaftsbeziehungen (Kanten). Der sich ergebende Pfad ist dann keine Sequenz von Wegpunkten, sondern eine Sequenz zu passierender Zellen. Da diese Zellen konvex sind, kann nach dem Erreichen der den Zielpunkt enthaltenden Zelle der Zielpunkt direkt angefahren werden. Eine solche konvexe Zelldekomposition kann analog zu Abschnitt 5.2 auf Polygonapproximationen der Hindernisse basieren, siehe Abbildung 5.8 nach [Top99, dCvO08].

An die Stelle einer exakten Konstruktion der Dekomposition kann aber auch eine Rasterung treten. Die Spielfeldfläche wird dabei regelmäßig in Zellen zerlegt, die entweder passierbar oder nicht passierbar sind. Der Einsatz eines rasterbasierten Verfahrens bietet sich an, wenn die realisierbare Auflösung ausreichend ist und das Überprüfen der Passierbarkeit einer Zelle die Dekomposition vereinfacht. Bei äquidistanter Rasterung ergibt sich ein Konflikt zwischen realisierbarer Auflösung und Speicher- bzw. Rechenaufwand. Dieser Konflikt motiviert eine adaptive Rasterung mit niedriger Auflösung in Gebieten gleicher Passierbarkeit. Die ad-

5. Pfadplanung

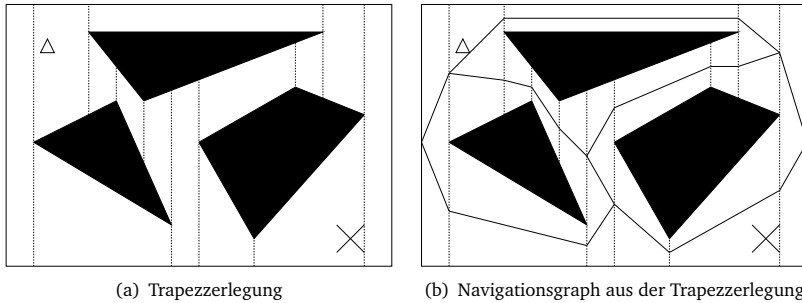


Abbildung 5.8.: Aufbau des Navigationsgraphen durch vertikale Trapezzerlegung

aktive Rasterung geschieht anhand eines Quadrees [Nil69, Eic07, dCv008]. Dieser wird benutzt, um die Spielfeldfläche in vollständig freie und vollständig gestörte rechteckige Zellen zu zerlegen. Teilweise gestörte Zellen werden dabei bis zu einer festgelegten Auflösungsgrenze geteilt. Damit ergeben sich die Knoten des Navigationsgraphen als vollständig freie Blätter des Baumes. Die Kanten des Navigationsgraphen werden aus der Nachbarschaft bezüglich der Polygonkanten dieser Knoten gebildet. Siehe Abbildung 5.9. Die Padsuche im Navigationsgraphen geschieht analog 5.2.

5.3.1. Eigenschaften einer Zelle

Die Zelle \mathcal{Z} ist ein Rechteck und somit Polygon der Eckpunkte $(Z_{NW}, Z_{NO}, Z_{SW}, Z_{SO})$ entsprechend Definition 15. Die Polygonkanten seien $\mathcal{K}_N = (Z_{NO}, Z_{NW})$, $\mathcal{K}_W = (Z_{NW}, Z_{SW})$, $\mathcal{K}_S = (Z_{SW}, Z_{SO})$ und $\mathcal{K}_O = (Z_{SO}, Z_{NO})$, siehe Abbildung 5.10.

Bei der Planung sollen die Hindernisse $\{\mathcal{O}_0, \dots, \mathcal{O}_{m-1}\}$ berücksichtigt werden. Dabei ist ein Hindernis \mathcal{O} eine Menge zu vermeidender Punkte. Haben das Hindernis \mathcal{O} und Zelle \mathcal{Z} gemeinsame Elemente, dann sei \mathcal{Z} durch \mathcal{O} gestört. Die Menge $\mathcal{Z}_\mathcal{O} = \mathcal{Z} \cap (\mathcal{O}_1 \cup \dots \cup \mathcal{O}_n)$ der Elemente, die die Zelle \mathcal{Z} mit der

5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition

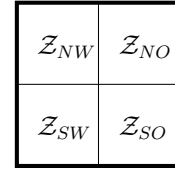
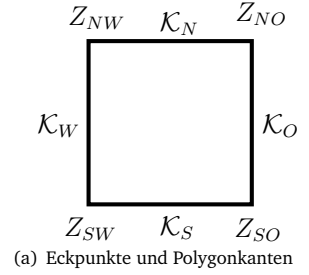
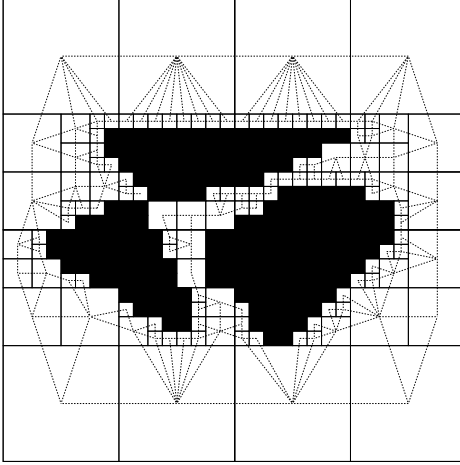


Abbildung 5.9.: Dekomposition der Spielfeldfläche in konvexe Zellen durch einen Quadtree. Navigationsgraph aus freien Blättern des Quadtree und geometrischer Blattnachbarschaft

Abbildung 5.10.: Eigenschaften einer Zelle \mathcal{Z}

Vereinigung der zu vermeidenden Hindernissen gemein hat, sei ein Maß für die Störung o einer Zelle:

$$o = \begin{cases} \text{Ungestört} & , \text{ wenn } \mathcal{Z}_O = \emptyset, \\ \text{Teilweise gestört} & , \text{ wenn } \emptyset \neq \mathcal{Z}_O \neq \mathcal{Z} \\ & \text{und Endauflösung nicht erreicht,} \\ \text{Vollständig gestört} & , \text{ sonst.} \end{cases} \quad (5.20)$$

Für den praktischen Einsatz wird \mathcal{Z}_O durch die maximale Störung durch ein einzelnes Hindernis angenähert. Dies vereinfacht die Implementation, führt aber zu fehlerhaften Erkennungen vollständig gestörter Zellen als teilweise gestörte, siehe Abbildung 5.11. Diese Näherung ist zulässig, da sie zu validen Dekompositionen führt. Allerdings werden unnötige Zellteilungen vorgenommen, weil manche vollständig beeinträchtigte Gebiete nicht erkannt werden.

5. Pfadplanung

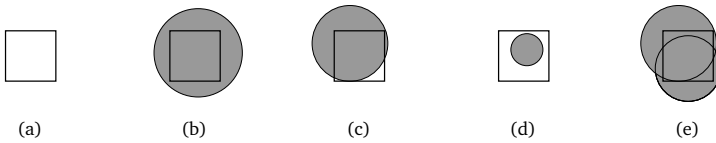


Abbildung 5.11.: Störungen einer Zelle: Vollständig frei (a) bzw. gestört (b), teilweise gestört (c) und (d), nicht als vollständig gestört erkannte vollständig gestörte Zelle (e).

Beim Aufbau des Quadtree werden teilweise gestörte Zellen entsprechend Abbildung 5.10(b) durch Halbierung entlang beider Achsen geteilt:

$$\begin{aligned} C : \{\mathcal{Z}\} &\rightarrow \{\mathcal{Z}\}^4 \\ \mathcal{Z} \mapsto C(\mathcal{Z}) &= (\mathcal{Z}_{NW}, \mathcal{Z}_{NO}, \mathcal{Z}_{SW}, \mathcal{Z}_{SO}). \end{aligned} \quad (5.21)$$

5.3.2. Adaptive Zelldekomposition durch Aufbau eines Quadtree

Die Zelldekomposition durch Aufbau eines Quadtree realisiert eine adaptive Rasterung des Spielfeldes. Im Gegensatz zu Zellaufteilungen durch kongruente Drei-, Vier- oder Sechsecke kann so eine hohe Auflösung in der Nähe von Hindernissen mit einer niedrigen Gesamtknotenzahl erreicht werden.

Der Quadtree G besteht als azyklischer Graph aus einer Menge von Knoten $\{n_i\}$ die von einer Menge von gerichteten Kanten $\{e_{i,j}\}$ verbunden sind. Dabei ist jeder Knoten des Baumes eine Zelle entsprechend Abschnitt 5.3.1. Der Knoten

$$p(n_q) := n_p \mid e_{p,q} \in \{e_{i,j}\} \quad (5.22)$$

sei der Elterknoten von n_q . Die Elemente der Menge der Nachfolger von n_p zu

$$c(n_p) := \{n_q \mid e_{p,q} \in \{e_{i,j}\}\} \quad (5.23)$$

seien Kinder des Knotens n_p . Hiermit ergeben sich die Kinder des Knotens n_p , die an der Polygonkante $\mathcal{K}_d(n_p)$ liegen zu

$$c_d(n_p) := \{n_q \in c(n_p) \mid |\mathcal{K}_d(n_q) \cap \mathcal{K}_d(n_p)| > 1\}. \quad (5.24)$$

Der Knoten ohne Elter ist die Wurzel des Baumes, alle Nicht-Wurzelknoten ohne Kinder seien Blätter.

5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition

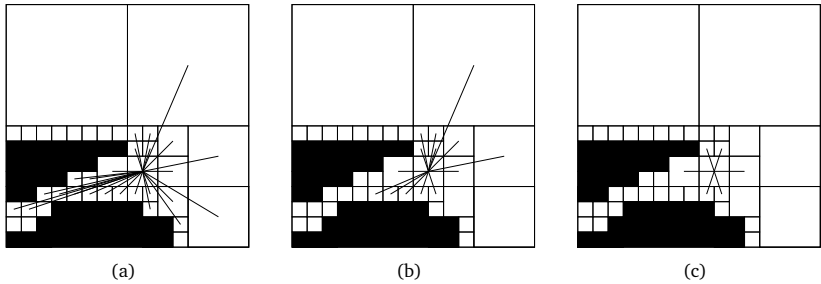


Abbildung 5.12.: Mögliche Varianten zur Definition der Blattnachbarschaft: Ist die Strecke zwischen zwei Blattzentren valide, dann sind die entsprechenden Blätter Nachbarn (a). Ist für beliebige Punkte aus zwei Blättern jede direkte Verbindung valide, dann sind die entsprechenden Blätter Nachbarn (b). Zwei benachbarte Blätter teilen sich eine Kante (c).

Dekomposition

Die Dekomposition geschieht, indem eine das Spielfeld repräsentierende Zelle als Wurzel des Baumes G angelegt wird. Solange G teilweise gestörte Blätter nach Gleichung (5.20) enthält, wird ein teilweise gestörtes Blatt n_p durch die Funktion (5.21) geteilt. Die erhaltenen Zellen werden als Kindknoten von n_p dem Baum hinzugefügt.

Zur Ermittlung, welches Blatt einen Prüfpunkt P enthält, wird von der Wurzel des Baumes ausgehend rekursiv geprüft, in welchem Kind P liegt. Hiermit ist es möglich, Start- sowie Zielknoten für die Planungsaufgabe zu bestimmen.

Bestimmung geometrischer Nachbarschaft

Abbildung 5.12 zeigt mögliche Interpretationen des Begriffes der Blattnachbarschaft. Bei der Variante 5.12(a) besteht Blattnachbarschaft, wenn die Verbindung der Blattzentren nicht durch ein Hindernis gestört ist. Diese Variante ist problematisch, da es Strecken zwischen Punkten von Nachbarblättern gibt, welche nicht valide sind. Deshalb ist dieser Nachbarschaftsbegriff nicht brauchbar im Sinne konvexer Zelldekomposition. In der Variante 5.12(b) besteht Blattnachbarschaft, wenn die Strecke zwischen beliebigen Punkten der Blätter valide ist. In der Variante 5.12(c) wird dieser Nachbarschaftsbegriff zugunsten einer einfachen Implementierung auf gemeinsam Kanten eingeengt.

5. Pfadplanung

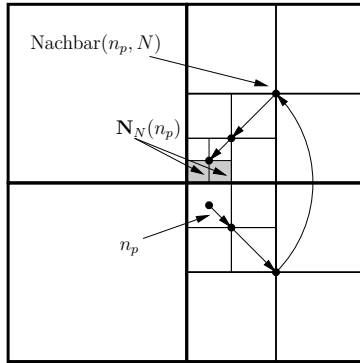


Abbildung 5.13.: Beispiel für die Suche der Nachbarschaft $N_N(n_p)$

Zur Nachbarschaft $N(n_p)$ des Blattes n_p mögen alle Blätter des Baumes gehören, die eine Kante von n_p mit mehr als einem Punkt berühren. Die Nachbarschaft eines Knotens n_p ergebe sich zu

$$\mathbf{N}(n_p) := \mathbf{N}_N(n_p) \cup \mathbf{N}_S(n_p) \cup \mathbf{N}_W(n_p) \cup \mathbf{N}_O(n_p) \quad (5.25)$$

durch die Kantennachbarschaften $N_d(u_p)$. Diese werden anhand Algorithmus 12 bestimmt, siehe Abbildung 5.13. In einem ersten Schritt wird hierfür durch Aufwärtstraversion der erste in Suchrichtung liegende Knoten nach Algorithmus 11 bestimmt.⁵ Dabei werden Richtungsinformationen über diese Traversion in einem Keller abgelegt. Diese werden in Algorithmus 12 ausgewertet, um die geometrische Nachbarschaft in einer Suchrichtung zu bestimmen. Hierfür werden die Blätter $L_d(n)$, die Nachfahren von n sind und die die Polygonkante $\mathcal{K}_d(n)$ mit mehr als einem Punkt berühren, zu

$$L_d(n) := \begin{cases} \{n\} & , c(n) = \emptyset \\ L_d(n_i) \cup L_d(n_j), n_i, n_j \in c_d(n), n_i \neq n_j & , \text{sonst.} \end{cases} \quad (5.26)$$

bestimmt.

⁵Eine analoge Umsetzung findet sich in [dCvO08].

5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition

Algorithmus 11: Nachbar

Eingabe : Knoten n , Suchrichtung d
1 Beginn
2 $t \leftarrow$ neuer Keller
3 **solange** $\exists p(n)$ **tue**
4 $s \leftarrow c_d(p(n)) \setminus \{n\}$
5 **wenn** $s \neq \emptyset$ **dann Ergebnis :** (s, t)
6 $z \leftarrow \begin{cases} NW & , \text{wenn } Z_{NW}(n) = Z_{NW}(p(n)), \\ NO & , \text{wenn } Z_{NO}(n) = Z_{NO}(p(n)), \\ SW & , \text{wenn } Z_{SW}(n) = Z_{SW}(p(n)), \\ SO & , \text{wenn } Z_{SO}(n) = Z_{SO}(p(n)) \end{cases}$
7 $e \leftarrow \begin{cases} \begin{cases} N, z = NO \\ N, z = NW \\ S, z = SW \\ S, z = SO \end{cases} & , d \in \{W, O\} \\ \begin{cases} O, z = NO \\ W, z = NW \\ W, z = SW \\ O, z = SO \end{cases} & , d \in \{N, S\} \end{cases}$
8 Kellere e in t ein
9 $n \leftarrow p(n)$
Ergebnis : $(\emptyset, \text{neuer Keller})$

Algorithmus 12: Nachbarn

Eingabe : Knoten n , Suchrichtung d
Ausgabe : N_d
1 Beginn
2 $(N, t) \leftarrow \text{Nachbar}(n)$
3 **wenn** $N = \emptyset$ **dann Ergebnis :** \emptyset
4 $n \leftarrow n_i \mid n_i \in N$
5 $g \leftarrow \begin{cases} S & , d = N \\ O & , d = W \\ N & , d = S \\ W & , d = O \end{cases}$
6 **solange** t nicht leer **tue**
7 **wenn** $c(n) \neq \emptyset$ **dann**
8 Kellere e aus t aus
9 $n \leftarrow n_q \mid n_q \in c_e(n) \cap c_g(n)$
10 **sonst**
Ergebnis : $\{n\}$
11 **wenn** $c(n) = \emptyset$ **dann Ergebnis :** $\{n\}$
Ergebnis : $L_g(n)$

5. Pfadplanung

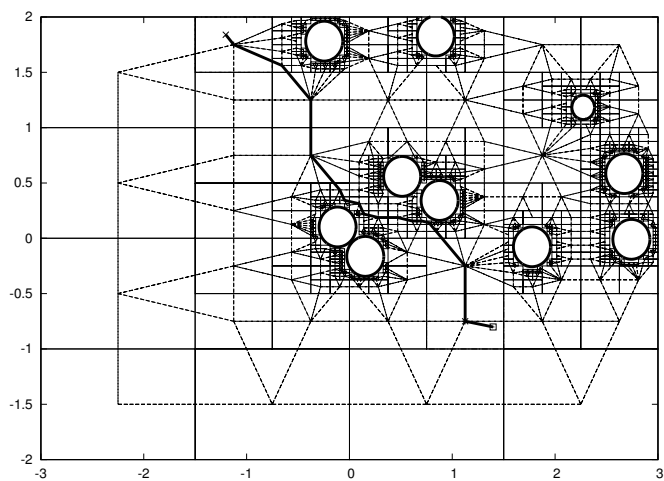
5.3.3. Vergleich zwischen adaptiver Zelldekomposition und konstruiertem Sichtbarkeitsgraph

Die Planungsaufgabe reduziert sich beim Verfahren der Quadtree-dekomposition auf die Beschreibung der Störung eines Rechtecks durch Hindernisse nach Gleichung (5.20). Die Beschreibung dieser Störung ist dabei vergleichsweise einfach auf verschiedene Klassen von Hindernissen wie Kreise, Ellipsen, Polygone oder Punktwolken anpassbar. Dabei darf eine Planungsaufgabe Hindernisse verschiedener Klassen umfassen. Gegenüber exakten geometrischen Verfahren, wie der Konstruktion eines Sichtbarkeitsgraphen⁶ ergibt sich hierdurch bei der Erweiterung um neue Hindernisklassen ein vergleichsweise niedriger Zuwachs an Komplexität.

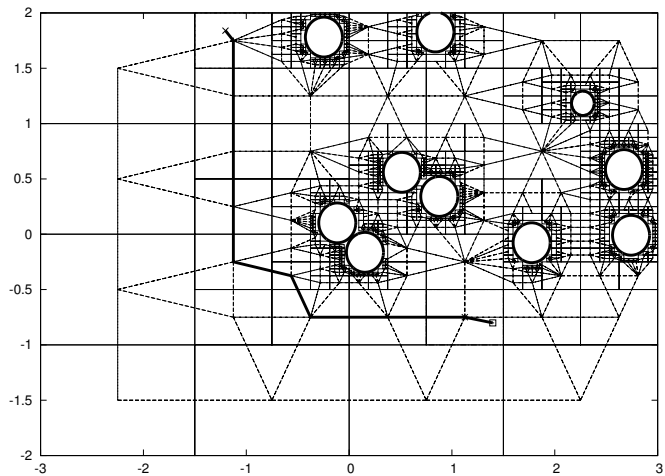
Weiterhin kann durch eine Modifikation der Transitionskosten der Abstand der generierten Pfade zu den Hindernissen beeinflusst werden. Wird zur Bestimmung der Transitionskosten benachbarter Blätter der euklidische Abstand der Blattzentren benutzt, dann ähneln die resultierenden Pfade denen eines Sichtbarkeitsgraphen. Werden die Transitionskosten jedoch um eine Wegpunktstrafe erhöht, dann bevorzugt die Planung größere Zellen, realisiert also Pfade mit zunehmendem Hindernisabstand, siehe Abbildung 5.14. Durch Veränderung eines einzelnen Parameters lässt sich somit die Charakteristik der generierten Pfade stark beeinflussen.

⁶Der Sichtbarkeitsgraph wurde in Abschnitt 5.2 für Approximationen der Hindernisse durch Polygone exakt aufgebaut.

5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition

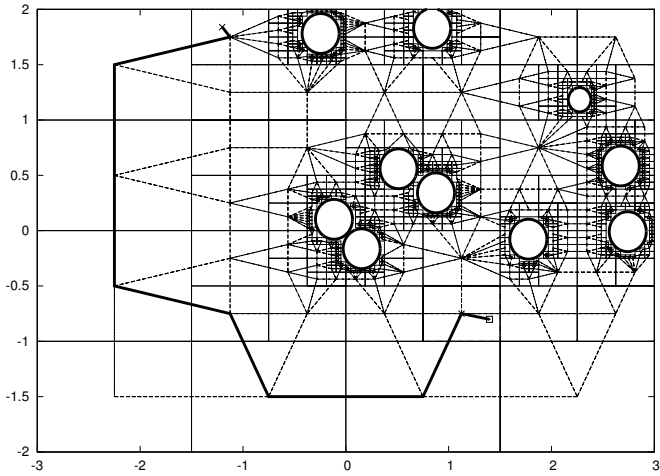


(a)

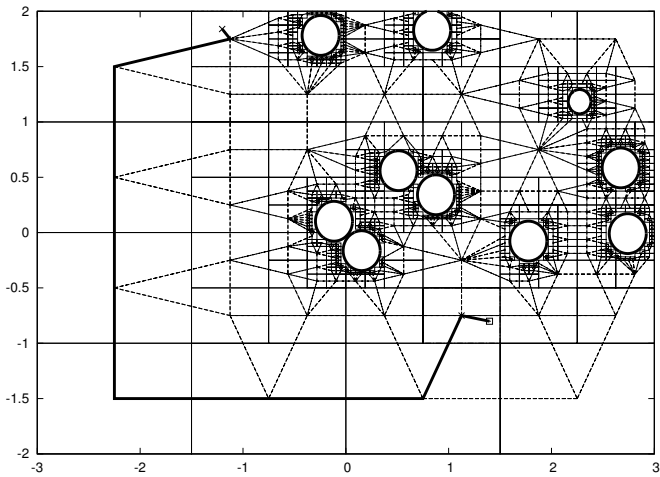


(b)

5. Pfadplanung



(c)



(d)

Abbildung 5.14.: Beispiel der Pfadsuche durch Quadtreedekomposition unter verschiedenen Varianten der Transitionskostendefinition: euklidischer Abstand der Blattzentren (a), euklidischer Abstand der Blattzentren mit zunehmender Wegpunktstrafe (b)-(d).

5.3. Aufbau des Navigationsgraphen durch adaptive Zelldekomposition

Die Tabellen 5.1 und 5.2 zeigen wichtige Parameter⁷ der beiden Verfahren. Dabei wird das Verhältnis aus Hindernisflächen und approximierten Hindernisflächen als Güte bezeichnet. Die Güte ist für kreisförmige Hindernisse dargestellt. Es zeigt sich, dass der Sichtbarkeitsgraph bei vergleichbarer Approximationsgüte weniger Knoten enthält, als der Navigationsgraph der Quadtree-dekomposition. Dabei weist der Sichtbarkeitsgraph allerdings eine quadratische Abhängigkeit von Knotenzahl und maximal durchzuführenden Kantentests auf.⁸⁹ Durch die Definition der Blattnachbarschaft anhand gemeinsamer Zellkanten werden beim Quadtree weniger Kantentests pro Knoten benötigt, im dargestellten Beispiel erscheint dieses Verhältnis über einen großen Bereich nahezu konstant. Im Gegensatz zur realisierten impliziten Formulierung des Sichtbarkeitsgraphen für die A*-Suche wird der Navigationsgraph durch Quadtree-dekomposition bisher vollständig aufgebaut, bevor die Pfadsuche beginnt. Durch eine implizite Formulierung des Verfahrens in dem Sinne, dass die Quadtree-dekomposition bei der Expansion eines Knotens des Navigationsgraphen nur in den relevanten Gebieten geschieht, wird eine wesentliche Effizienzsteigerung des Verfahrens erwartet. In den aktuellen Implementierungen hat die Pfadsuche anhand des Sichtbarkeitsgraphen für Pentagone einen Geschwindigkeitsvorteil größer als Faktor 15 gegenüber der Quadtree-dekomposition bei vergleichbarer Approximationsgüte. Deshalb wird diese Variante favorisiert.

Auflösung des [mm]	Knoten des Quadtree	Blätter des Quadtree	Knoten des Navi- gations- graphen	Kantentests	Güte	Güte / Kan- tentests [‰]
250	305	229	137	632	0.44	0.70
125	633	475	255	1222	0.65	0.53
60	1317	988	430	2128	0.79	0.37
30	2709	2032	822	4080	0.90	0.22
15	5493	4120	1544	7700	0.96	0.12

Tabelle 5.1.: Parameter des Navigationsgraphen aus der Quadtree-dekomposition

⁷Die Parameter des Sichtbarkeitsgraphen sind analytisch, die Parameter der Quadtree-dekomposition aus einem Beispiel bestimmt.

⁸Die Anzahl zu überprüfender Kanten kann durch den Einsatz eines reduzierten Sichtbarkeitsgraphen [CLH⁺05] stark reduziert werden.

⁹Als Folge der impliziten Formulierung des Sichtbarkeitsgraphen für die Pfadsuche mittels A*-Algorithmus ist die tatsächliche Anzahl durchgeführter Kantentests in der Praxis wesentlich niedriger als dieses Maximum

5. Pfadplanung

Eckpunktanzahl	Knoten	Kantentests	Güte	Güte / Kantentests [%]
3	32	992	0.60	0.61
4	42	1722	0.79	0.46
5	52	2652	0.86	0.33
6	62	3782	0.91	0.24
7	72	5112	0.93	0.18
8	82	6642	0.95	0.14
9	92	8372	0.96	0.11

Tabelle 5.2.: Parameter des Sichtbarkeitsgraphen aus Polygonen

5.4. Anwendung der Pfadplanung

5.4.1. Nachbearbeitung eines Pfades

Nachdem ein Pfad als Sequenz abzufahrender Wegpunkte (W_1, \dots, W_n) erstellt wurde, um den Roboter der Position R zur Zielposition T zu bewegen, soll diesem Pfad möglichst zeitsparend gefolgt werden. Hierfür steht der Lageregelkreis zu Verfügung. Der eingesetzte Regler berücksichtigt die Beschleunigungsfähigkeit des Roboters, so dass das Ziel ohne Überspringen erreicht werden kann. Die Forderung, das Ziel ohne Überspringen zu erreichen, ergibt sich aus dem Vorhandensein von Hindernissen, wird sie verletzt, drohen Kollisionen. Hieraus resultiert je nach Wegpunktdichte eine stark reduzierte Dynamik.

Deshalb wird ein geometrischer Ansatz zur Steigerung der Dynamik für das Abfahren eines Pfades durch gezieltes Ausnutzen von Überspringen verwendet, siehe Abbildung 5.15.

Ziel ist es, einen Ersatzwegpunkt W_1''' auf dem Strahl von R durch W_1 zu finden, für den die Geschwindigkeit des Roboters beim Erreichen von W_1 maximal ist. Hierzu werden die Spurparameter $\lambda_2, \dots, \lambda_n$ durch Projektion von $\vec{w}_2 - \vec{r}, \dots, \vec{w}_n - \vec{r}$ auf den Strahl bestimmt. Das erste Maximum dieser Spurparameter führt zum Ersatzwegpunkt W_1' , siehe Abbildung 5.15(a). In einem zweiten Schritt wird die Strecke von R nach W_1' auf Kollisionen bezüglich der Hindernisse \mathcal{O}_i untersucht. Dabei wird W_1'' als Ersatzwegpunkt des maximalen validen Spurparameters für den Strahl von R durch W_1 bestimmt (Abbildung 5.15(c)). Im letzten Schritt wird $W_1''' \in RW_1'$ bestimmt, indem der maximale Spurparameter gefunden wird, für den $W_1'''W_2$ kollisionsfrei ist (Abbildung 5.15(d)).

Der Ersatzwegpunkt W_1''' wird als Führungsgröße des Lageregelkreises benutzt. Das Überspringen des Reglers bleibt auf das Dreieck W_1, W_1'', W_2 beschränkt. Hierdurch ist Kollisionsfreiheit weiterhin sichergestellt.

5.4.2. Realisierte Manöver

Die bisher dargestellten Verfahren der digitalen Bildverarbeitung und der modellbasierten Lageregelung bilden die Grundlage der Umgebungsmodule. In den hierdurch bereitgestellten realen oder simulierten Umgebungen können Softwaremodule der Mannschaftsstrategie zum autonomen Fußballspiel gegeneinander antreten. Das erstellte Modul der Mannschaftsstrategie realisiert eine kaskadierbare reaktive Struktur. In dieser Struktur werden einzelne Teilverhalten durch *Zustände* endlicher Automaten modelliert. Die beiden elementaren Zustände sind ein Schussmanöver und ein defensives Manöver. Diese beiden Teilverhalten basieren auf den im Folgenden dargestellten geometrischen Überlegungen.

Torschuss Eine einfache Vorbereitung eines Torschusses durch geometrische Konstruktion ist in Abbildung 5.16 dargestellt. Dabei geschieht die Approximation des Balles durch ein Polygon in Abhängigkeit von der Ausrichtung des Spielers zum Ball bzw. von deren Zeitverlauf. Die Zielposition T des Spielers liegt auf dem Strahl aus dem Schussziel G durch die Ballposition B . Sie ist der dem Schussziel abgewandte Schnittpunkt des Strahls mit dem Polygon. Mit Hilfe der Pfadplanung wird ein Pfad von der Spielerposition R zu T erstellt, um Kollisionen mit den Hindernissen O_i zu vermeiden. Diesem Pfad wird durch gezieltes Überspringen zeitsparend gefolgt. Eine Erweiterung dieses Verfahrens für eine dynamische Umgebung wurde realisiert, der praktische Einsatz wird jedoch durch die Geschwindigkeitsabhängigkeit des Verfahrens erschwert.

5. Pfadplanung

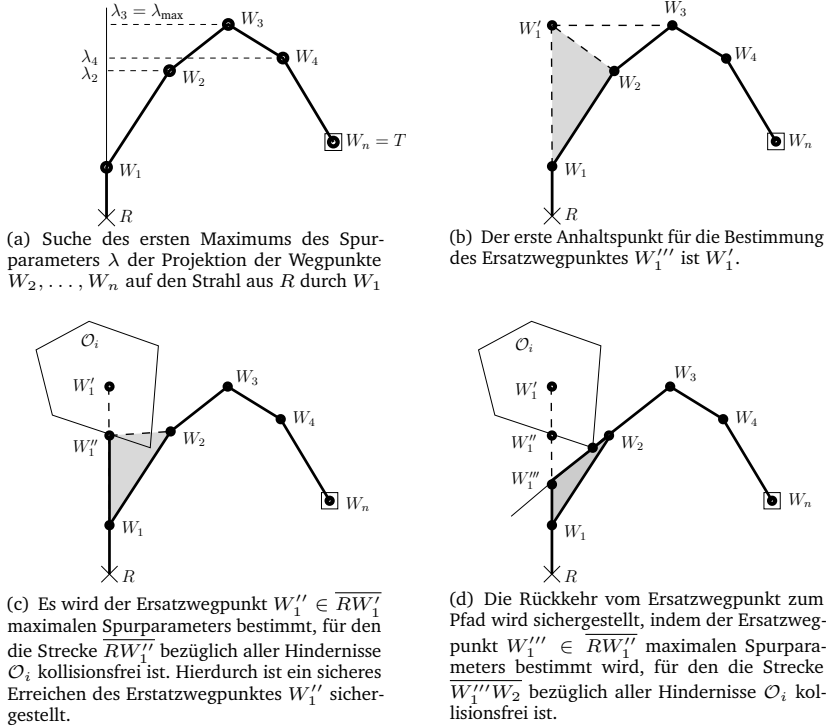


Abbildung 5.15.: Bestimmung des Ersatzwegpunktes W'_1 , für maximale Geschwindigkeit bei W_1 .

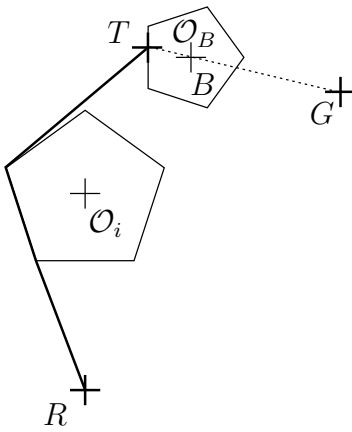
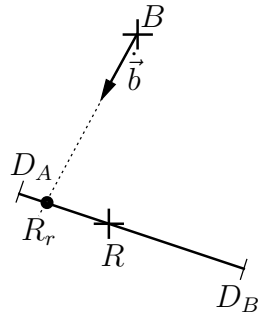
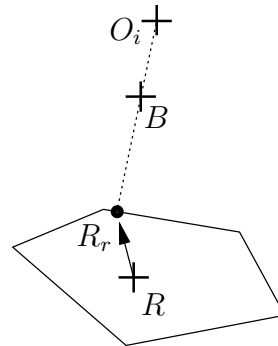


Abbildung 5.16.: Torschuss



(a) Linienverteidigung



(b) Alternative Verteidigung

Abbildung 5.17.: Verteidigung

5. Pfadplanung

Linienverteidigung Eine Variante zur Umsetzung eines Verteidigungsverhaltens ist in Abbildung 5.17(a) dargestellt. Die Verteidigungsstrecke $\overline{D_A D_B}$ sei von strategischer Bedeutung. Gibt es einen Schnittpunkt R_r des Strahls aus der Ballposition B in Richtung der Ballgeschwindigkeit \vec{b} mit dieser Strecke, so ist R_r die Sollposition des Spielers. Problematisch an dieser Variante sind die Abhängigkeit von der Ballgeschwindigkeit sowie die eingeschränkte Querdynamik des aktuellen Spielers. Das erste Problem kann durch Verwendung eines besser geeigneten Aktivierungskriteriums gelöst werden, indem beispielsweise Informationen über den den Ball nächstgelegenen Gegner ausgewertet werden. Die zweite Schwäche könnte durch das Auswechseln des Verteidigungsprimitivs abgemildert werden. Würde die Bestimmung der Sollposition nicht durch Schneiden des Ballstrahls mit einer Verteidigungsstrecke, sondern mit einem Verteidigungspolygon erfolgen, dann könnte die Beschleunigungsfähigkeit des Spielers besser genutzt werden. Siehe Abbildung 5.17(b).

6. Zusammenfassung und Ausblick

Ziel dieser Arbeit war die Erstellung von Software zur Steuerung einer Mannschaft der RoboCup Small Size League. Die Grundstruktur dieser Software besteht aus einem Modul der Umgebung und Modulen der Strategie. Ein Softwaremodul der Umgebung bietet Methoden zur Wahrnehmung und zur Beeinflussung einer realen oder simulierten Spielumgebung, in der zwei Mannschaften im autonomen Fußballspiel konkurrieren können. Der Schwerpunkt dieser Arbeit lag auf der Erarbeitung wichtiger Teile dieser Umgebungen. Ein Modul der Strategie wird durch Ereignisse über den Zustand der Umgebung informiert. Es agiert in der Umgebung durch das Senden von Steuerbefehlen an diese Umgebung. Module der Mannschaftsstrategie können in einer Spielumgebung bereits einfache Teilaufgaben wie die kollisionsfreie Bewegung eines Spielers zu einer Solllage, einen Torschuss oder rudimentäre Verteidigungen lösen. Die vorliegende Arbeit stellt einen Beitrag zur Lageregelung eines technischen Systems mit nicht-konvexem Zustandsraum und konvexer Beschränkung der Steuergröße dar. Hierfür wurden Verfahren der Lagebestimmung durch digitale Bildverarbeitung angewendet und angepasst. Der Schwerpunkt des Reglerentwurfs lag bei der Modellbildung, hierbei wurde eine allgemeine Beschreibung omnidirektionaler Antriebssysteme mit Grabowieckirädern erarbeitet. Auf Grundlage dieses Modells wurden einfache Geschwindigkeits- und Lageregler realisiert, welche die konvexe Beschränkung der Steuergröße berücksichtigen. Der Aspekt des nicht-konvexen Zustandsraums wurde durch Anwendung von Planungsalgorithmen zur Generierung der Führungsgröße der Lageregelung berücksichtigt.

In den Kapiteln 2 und 3 wurde die Detektion der Spieler in einer realen Spielumgebung durch digitale Bildverarbeitung behandelt. Die Spieler sind durch Farbmarkierungen gekennzeichnet. Das entscheidende Problem an dieser Teilaufgabe ist die Bildauswertung unter Echtzeitanforderungen. Es wurde mit einem segmentbasierten Bildauswertungsverfahren nach Bruce, Balch und Veloso [BBV00] gelöst. Mit diesem Verfahren werden Bildbereiche gleicher Zugehörigkeit zu einer Farbkategorie detektiert. Die Besonderheit an diesem Verfahren ist, dass die Segmentierung eines Bildes in zwei Schritten erfolgt. Im ersten Schritt wird eine Lauflängenkodierung klassifizierter Bildelemente durchgeführt und im zweiten Schritt werden benachbarte Gruppen von Bildelementen mit gleicher Farbzugehörigkeit zu Segmenten vereinigt. Durch Unterschiede des

6. Zusammenfassung und Ausblick

verwendeten Bildaufnahmesystems begründet, erfolgt eine Modifikation der Farbklassifikation. Während diese in [BBV00] anhand einhüllender Quader realisiert wird, geschieht sie in der vorliegenden Arbeit durch einen statischen diskreten Voronoiklassifikator. Varianten dieses Klassifikators zur Unterscheidung von acht Farbklassen wurden durch den Satz von Bayes, die Mahalanobisdistanz und den elementweisen Minimalabstand zu einer Stichprobe erstellt. Im praktischen Einsatz haben sich diese Varianten als ausreichend robust erwiesen. Der Klassifikator wird aus Stichproben der einzelnen Farbklassen erzeugt. Diese Stichproben werden durch den Nutzer bereitgestellt. Dadurch dass diese Stichproben nur an wenigen Messorten erhoben werden, weisen sie jedoch eine unvollständige Abdeckung der zugrunde liegenden Verteilungen auf. Interessant erscheint deshalb die Untersuchung von Möglichkeiten, den Klassifikator im Spielbetrieb zu adaptieren, um diese Abdeckung zu verbessern. In gleicher Weise könnte die vorgenommene lokale Helligkeitsanpassung im Spielbetrieb adaptiert werden. Es wird erwartet, dass eine Kombination dieser Maßnahmen zu einer erheblichen Steigerung der Robustheit des Systems führen wird. Bei der Umsetzung dieser Maßnahmen werden die größten Herausforderungen in der Sicherstellung von Stabilität und Echtzeitfähigkeit erwartet.

Es wurde versucht, durch Verfahren der Clusteranalyse und der Mustererkennung eine autonome Partitionierung der im Videosignal realisierten Farbwerte entsprechend den vorkommenden Farbklassen vorzunehmen. Diese autonome Partitionierung gelang mit den untersuchten Methoden nicht. Diese Teilaufgabe ist zwar für die Objektlokalisierung im RoboCup weniger wichtig, als Fragestellung des unüberwachten Lernens könnten Fortschritte bei diesem Problem jedoch direkt für Fragen der Spielstrategie relevant werden.

Die detektierten Segmente werden zur Bestimmung der Objektlagen durch die Helmert-Transformation ausgewertet. Dieses Verfahren der Ausgleichsrechnung wurde durch die Wichtung von Einzelfehlern für den Einsatz im Kontext digitaler Bildverarbeitung angepasst. Hierdurch steht bei der Ermittlung der Spielerlagen Zusatzwissen über die zu erwartende Objektgeometrie zur Verfügung. Ein synthetisches Experiment zeigte, dass dieses Zusatzwissen zu einer Steigerung der Robustheit des Verfahrens bezüglich Fehlklassifikationen führt. Vor der Anwendung der Helmert-Transformation werden die Schwerpunkte der Markierungen im Spielfeldkoordinatensystem bestimmt, hierfür wurde eine einfache Kalibrierung des Systems vorgenommen. Es wird erwartet, dass die Messgenauigkeit durch Verwendung verbesserter Methoden der Kamerakalibrierung gesteigert werden kann.

Der zweite wichtige Aspekt des Moduls der Umgebung betrifft die Ansteuerung der Spieler. Ein Modul der Strategie soll den Zustand der Umgebung durch das

Senden von Lagebefehlen beeinflussen können. Die Lageregelung selbst ist ein Teil der realen oder simulierten Spielumgebung. Da die simulierte Umgebung ein wichtiges Hilfsmittel für die Entwicklung von Mannschaftsstrategien ist, wurde eine hohe Simulationsgüte angestrebt. Hierfür wurde im Kapitel 4 ein detailliertes Modell für omnidirektional angetriebene Systeme beliebiger Radanordnungen erarbeitet. Diese Systeme wurden als Mehrkörpersysteme modelliert, um die Begrenzungen der Beschleunigungsfähigkeit durch die Haftreibungsgrenze und elektrisch realisierbare Antriebsmomente zu beschreiben. Die Ermittlung der Beschleunigungsfähigkeit wurde in ein lineares Optimierungsproblem überführt und durch das Simplexverfahren für Systeme der Radanzahl größer zwei gelöst. Diese allgemeine Formulierung wurde gewählt, um das Modell auch für andere RoboCup-Teams verwertbar zu machen. Ein weiteres Ergebnis dieser Modellierung ist die explizite Darstellung der mechanischen Verkopplung des Systems sowie deren Invertierung als Teillösung des linearen Optimierungsproblems. Obwohl das Modell einen hohen Detailierungsgrad aufweist, ist es unter den Echtzeitanforderungen im Reglerbetrieb auswertbar. Das erarbeitete Modell ist eine Verallgemeinerung des Modells nach Landsmann [Lan05] für beliebige Radanordnungen, welches gut experimentell bestätigt ist. Im Vergleich zu etablierten Modellen [CGD⁺01, RF06, LZWW08] wird hier die räumliche Struktur des Roboters beachtet, woraus eine potentiell genauere Bestimmung der Beschleunigungsfähigkeit resultiert. Obwohl die mit Hilfe des Modells durchgeführten Simulationen gute Übereinstimmung mit dem Modell nach [Lan05] zeigten, bleibt die direkte Verifizierung durch Messungen eine offene Fragestellung für zukünftige Arbeiten. Interessant erscheint insbesondere die Frage, ob die Regelgüte konkurrierender Systeme durch den Einsatz des vorliegenden Modells verbessert werden kann.

Aufbauend auf dem Modell wurden ein Geschwindigkeits- und zwei Lageregler erarbeitet. Der Geschwindigkeitsregler korrigiert Geschwindigkeitsfehler an der Beschleunigungsgrenze. Beide Lageregler verwenden eine Ansatzfunktion für den Zeitverlauf der Beschleunigung, die jeweiligen Parameter werden mit Hilfe des Modells bestimmt. In einer Variante wurde eine lineare Funktion als Ansatz des Beschleunigungsverlaufes verwendet. Diese Variante zeichnet sich durch das Ausregeln der Solllage ohne Überspringen aus. In einer zweiten Variante wurde ein stückweise konstanter Beschleunigungsverlauf gewählt. Hiermit wurde eine bessere Dynamik des Systems erzielt, jedoch trat in bestimmten Betriebsfällen Überspringen auf. Zur Begrenzung des maximalen Überspringens auf eine vorgegebene Toleranz wurde ein Sicherheitsfaktor für die angenommene Bremsbeschleunigung eingeführt. Dieser wurde durch Auswertung einer Menge von Testfällen mit einem Suchverfahren bestimmt. Die bereitgestellten Regler bilden einen Teil der Schnittstelle zwischen Umgebung und Modul der

6. Zusammenfassung und Ausblick

Mannschaftsstrategie. Sie wurden für das Simulationsmodul implementiert und sollen auf dem noch nicht fertiggestellten Prototypen zum Einsatz kommen. Die positiven Erfahrungen mit dem Lageregler nach [Lan05] lassen eine gute Übereinstimmung von simuliertem und realem Spieler vermuten. Eine Konsequenz hieraus wäre die Möglichkeit, die Ergebnisse der Bildauswertung, insbesondere im Hinblick auf die Rekonstruktion der Geschwindigkeit, durch den Einsatz von Zustandsbeobachtern zu verbessern. Für eine Realisierung des Reglers für einen realen Spieler sind die Teilaufgabe der technischen Umsetzung auf der Zielplattform mit eingeschränkten Ressourcen sowie die Teilaufgaben der Gestaltung von Messsystemen und der modellgestützten Messwertaufbereitung zu lösen. Eine wichtige offene Fragestellung ist die Verbesserung des Lagereglers selbst, besonders in der Variante der stückweise konstanten Beschleunigung. Es ist anzustreben, die pragmatische Umsetzung mit Hilfe des empirisch bestimmten Sicherheitsfaktors durch eine zeitoptimale Steuerung abzulösen. Hierbei werden jedoch die Ressourcen der Zielplattform ein entscheidendes Hindernis darstellen.

Im Modul der Mannschaftsstrategie wurde ein hybrid deliberativ-reaktiver Ansatz verfolgt. Einzelspieler- und Mannschaftsverhalten wurden durch eine reaktive Struktur kaskadierbarer endlicher Automaten modelliert. In dieser Struktur wurde die Teilaufgabe der Überführung eines Spielers in eine Solllage unter Vermeidung von Kollisionen durch kombinatorische Planung gelöst. Hierfür wurde in Kapitel 5 der A*-Algorithmus zur Suche in einem Navigationsgraphen angewendet. Es wurden zwei Varianten zur Erzeugung des Navigationsgraphen implementiert. Eine Variante ist die Erzeugung eines Sichtbarkeitsgraphen aus Approximationen der Hindernisse durch konvexe Polygone. Bedingt durch die kleine Problemgröße des Planungsproblems und seine implizite Formulierung konnten Planungsgeschwindigkeiten von $> 10^3$ Pfaden pro Sekunde erzielt werden, ohne optimierte Algorithmen zur Konstruktion des Sichtbarkeitsgraphen anzuwenden. Eine zweite Variante zur Erzeugung des Navigationsgraphen wurde mit einer adaptiven Rasterung realisiert. Hierbei wurde das Spielfeld durch einen Quadtree in passierbare und unpassierbare Zellen zerlegt. Der Navigationsgraph wurde hierbei durch die Auswertung geometrischer Nachbarschaftsbeziehungen zwischen den Blättern des Quadtree generiert. Bei diesem Verfahren konnte die Charakteristik der ermittelten Pfade durch einen einzigen Parameter stark beeinflusst werden. Für die Pfadsuche ergab sich aber bei vergleichbarer Approximationsgüte ein wesentlich erhöhter Zeitbedarf.

Die bisher erstellten Verhaltensbausteine basieren auf der Analyse elementarer Teilprobleme. Die Synthese von robustem und effektivem Spieler- und Mannschaftsverhalten aus diesen Bausteinen wird jedoch als erhebliche Heraus-

forderung angesehen. Als vielversprechende Alternative erscheint der Einsatz dieser Bausteine zusammen mit Verfahren des bestärkenden Lernens. Bei diesem Ansatz würden „gut“ analytisch beschreibbare Teillösungen als Zustände bzw. Aktionen implementiert. Der „schlecht“ analytisch beschreibbare effektive Einsatz dieser Bausteine würde jedoch vom Strategiemodul erlernt werden. Diese Eigenschaft ist vor allem in Hinblick auf wechselnde Gegner relevant. Das Ziel des RoboCup-Projektes der Fachgebiete Rechneranwendung im Maschinenbau und Technische Mechanik ist es, erfolgreich an einem Wettbewerb in der Small Size League teilzunehmen. Hierfür ist es notwendig, in einer sich schnell verändernden Umgebung koordinierte und präzise Handlungen ohne menschlichen Eingriff durchzuführen. Die anschauliche Aufgabe des Fußballspiels autonomer technischer Systeme hält hiermit eine Fülle anspruchsvoller und für praktische Anwendungen relevanter Teilaufgaben bereit. In dieser Arbeit wurden wichtige Teilaufgaben zur Mannschaftssteuerung gelöst. Hiermit allein ist noch keine Teilnahme an einem Wettkampf denkbar, sie bieten aber eine solide Grundlage für die Entwicklung leistungsfähiger Spielstrategien.

A. Verwendung des Computeralgebrasystems Maxima

Zur Formulierung des Mechanikmodells wurde das Softwarepaket Maxima [max09] genutzt. Maxima ist ein freies Computeralgebrasystem zur Manipulation symbolischer Ausdrücke. Am Beispiel der zeitlichen Ableitung des Radimpulses soll die Verwendung von Maxima zur Lösung wichtiger Zwischenschritte im Rahmen vorliegender Arbeit skizziert werden. Das Modell wird durch Ausdrücke und Operationen vom Nutzer beschrieben, Maxima wertet diese Eingabe symbolisch aus. Die Ausgaben des Programmes werden dann bei der weiteren Ausgestaltung des Modells verwendet.

Eingabe

```
/*Definition eigener Funktionen*/
V3(a,b,c):=matrix([a],[b],[c])$
RotZ(a):=matrix([ cos(a), sin(a), 0      ],
                 [-sin(a), cos(a), 0      ],
                 [ 0,      0,      1      ])
)$

/*Koordinatensystem des Roboters im Roboterschwerpunkt*/
/*Position des Roboterschwerpunktes
   im globalen Koordinatensystem*/
/*Definition eines Vektors mit zeitabhaengigen
   und zeitunabhaengigen Komponenten*/
gR:V3(x(t),y(t),z)$
/*Basis des Robotersystems im globalen Koordinatensystem*/
/*Definition einer zeitabhaengigen Matrix*/
BR:RotZ(Psi(t))$
/*Inverse der Basis des Robotersystems
   im globalen Koordinatensystem*/
/*Symbolische Manipulation*/
BRi:trigsimp(invert(BR))$
```

A. Verwendung des Computeralgebrasystems Maxima

```
/*Position der Radnabe im Robotersystem*/
lW:V3(lWx,lWy,lWz)$

/*Position der Radnabe im globalen System*/
disp("Position der Radnabe, gW=")$
/*Vektoraddition und Matrizenmultiplikation*/
gW:gR+BRi.lW;

disp("Globale Nabengeschwindigkeit, dgW=")$
/*Zeitliche Ableitung*/
dgW:diff(gW,t);

/*Impuls des Rades*/
disp("Impuls des Rades")$
gPW:mW * dgW;
disp("Impulsaenderung des Rades")$
dgPW:diff(gPW,t);
```

Ausgabe

```
Position der Radnabe, gW=
(%o10) matrix(
  [-sin(Psi(t))*lWy+cos(Psi(t))*lWx+x(t)],
  [cos(Psi(t))*lWy+sin(Psi(t))*lWx+y(t)],
  [lWz+z]
)

Globale Nabengeschwindigkeit, dgW=
(%o12) matrix(
  [-cos(Psi(t))*(diff(Psi(t),t,1))*lWy-sin(Psi(t))
    *(diff(Psi(t),t,1))*lWx +diff(x(t),t,1)],
  [-sin(Psi(t))*(diff(Psi(t),t,1))*lWy+cos(Psi(t))
    *(diff(Psi(t),t,1))*lWx +diff(y(t),t,1)],
  [0]
)

Impuls des Rades
(%o14) matrix(
  [(-cos(Psi(t))*(diff(Psi(t),t,1))*lWy-sin(Psi(t))
```

```

        *('diff(Psi(t),t,1))*lWx +'diff(x(t),t,1))*mW],
[(-sin(Psi(t))*('diff(Psi(t),t,1))*lWy+cos(Psi(t))
        *('diff(Psi(t),t,1))*lWx +'diff(y(t),t,1))*mW],
[0]
)

```

Impulsaenderung des Rades

```

(%o16) matrix(
[(-cos(Psi(t))*('diff(Psi(t),t,2))*lWy+sin(Psi(t))
        *('diff(Psi(t),t,1))^2*lWy
        -sin(Psi(t))*('diff(Psi(t),t,2))*lWx-cos(Psi(t))
        *('diff(Psi(t),t,1))^2*lWx
        +'diff(x(t),t,2))*mW],
[(-sin(Psi(t))*('diff(Psi(t),t,2))*lWy-cos(Psi(t))
        *('diff(Psi(t),t,1))^2*lWy
        +cos(Psi(t))*('diff(Psi(t),t,2))*lWx-sin(Psi(t))
        *('diff(Psi(t),t,1))^2*lWx
        +'diff(y(t),t,2))*mW],
[0]
)

```


B. Matrizen des Mechanikmodells

Verallgemeinerte Trägheitsmatrix der gesteuerten Koordinaten

$$\mathbf{B} = \begin{pmatrix} b_{1,1} & 0 & b_{1,3} \\ 0 & b_{1,1} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} \quad (\text{B.1})$$

$$b_{1,1} = m_B + \sum_i \left[m_{Hi} + m_{Si} \right] \quad (\text{B.2})$$

$$b_{1,3} = -m_B \underline{b}_y - \sum_i \left[m_{Hi} \underline{h}_{iy} + m_{Si} \underline{s}_{iy} \right] \quad (\text{B.3})$$

$$b_{2,3} = m_B \underline{b}_x + \sum_i \left[m_{Hi} \underline{h}_{ix} + m_{Si} \underline{s}_{ix} \right] \quad (\text{B.4})$$

$$b_{3,1} = \sum_i \left[m_{Hi} (\underline{b}_y - \underline{h}_{iy}) + m_{Si} (\underline{b}_y - \underline{s}_{iy}) \right] \quad (\text{B.5})$$

$$b_{3,2} = \sum_i \left[m_{Hi} (\underline{h}_{ix} - \underline{b}_x) + m_{Si} (\underline{s}_{ix} - \underline{b}_x) \right] \quad (\text{B.6})$$

$$\begin{aligned} b_{3,3} = \underline{\underline{J}}_{Bzz} + \sum_i \left[\underline{\underline{J}}_{Hizz} + \underline{\underline{J}}_{Sizz} + m_{Hi} \left(\underline{h}_{iy} (\underline{h}_{iy} - \underline{b}_y) + \underline{h}_{ix} (\underline{h}_{ix} - \underline{b}_x) \right) \right. \\ \left. + m_{Si} \left(\underline{s}_{iy} (\underline{s}_{iy} - \underline{b}_y) + \underline{s}_{ix} (\underline{s}_{ix} - \underline{b}_x) \right) \right] \quad (\text{B.7}) \end{aligned}$$

Verallgemeinerte Matrix der Corioliskräfte für die gesteuerten Koordinaten

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & b_{2,3} \\ 0 & 0 & -b_{1,3} \\ 0 & 0 & \sum_i \left[m_{Hi} (\underline{b}_y \underline{h}_{ix} - \underline{b}_x \underline{h}_{iy}) + m_{Si} (\underline{b}_y \underline{s}_{ix} - \underline{b}_x \underline{s}_{iy}) \right] \end{pmatrix} \quad (\text{B.8})$$

B. Matrizen des Mechanikmodells

Verallgemeinerte Matrix der Zwangskräfte für die gesteuerten Koordinaten

$$\mathbf{Z} = \begin{pmatrix} \cos(\varphi_1) & \dots & \cos(\varphi_n) \\ \sin(\varphi_1) & \dots & \sin(\varphi_n) \\ z_{3,1} & \dots & z_{3,n} \end{pmatrix} \text{diag}(k_1, \dots, k_n), \quad (\text{B.9})$$

$$z_{3,i} = \cos(\varphi_i)(\underline{b}_y - \underline{h}_{iy}) + \sin(\varphi_i)(\underline{h}_{ix} - \underline{b}_x) \quad (\text{B.10})$$

Verallgemeinerte Matrix der Antriebskräfte für die gesteuerten Koordinaten

$$\mathbf{D} = \begin{pmatrix} -\sin(\varphi_1) & \dots & -\sin(\varphi_n) \\ \cos(\varphi_1) & \dots & \cos(\varphi_n) \\ d_{3,1} & \dots & d_{3,n} \end{pmatrix} \quad (\text{B.11})$$

$$d_{3,i} = \sin(\varphi_i)(\underline{h}_{iy} - \underline{b}_y) + \cos(\varphi_i)(\underline{h}_{ix} - \underline{b}_x) \quad (\text{B.12})$$

Einfluss der Erdbeschleunigung auf die fixierten Koordinaten

$$\vec{\mathbf{G}} = \begin{pmatrix} \sum_i [m_{Hi}(\underline{h}_{iy} - \underline{b}_y) + m_{Si}(\underline{s}_{iy} - \underline{b}_y)] \\ \sum_i [m_{Hi}(\underline{b}_x - \underline{h}_{ix}) + m_{Si}(\underline{b}_x - \underline{s}_{ix})] \\ \sum_i [m_{Hi} + m_{Si}] + m_B \end{pmatrix} g \quad (\text{B.13})$$

Verallgemeinerte Trägheitsmatrix der fixierten Koordinaten

$$\mathbf{N} = \begin{pmatrix} n_{1,1} & n_{1,2} & n_{1,3} \\ n_{2,1} & n_{2,2} & n_{2,3} \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{B.14})$$

$$n_{1,1} = \sum_i \left[\frac{1}{2r_i} \left(-\gamma_i \underline{\underline{J}}_{S_{ixx}} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) - \underline{\underline{J}}_{H_{ixx}} \sin(2\varphi_i) \right) \right] \quad (\text{B.15})$$

$$\begin{aligned} n_{1,2} = \sum_i & \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} (\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i)) \right. \right. \\ & \left. \left. + \underline{\underline{J}}_{H_{ixx}} (\cos(2\varphi_i) + 1) \right) + m_{Hi}(\underline{h}_{iz} - \underline{b}_z) + m_{Si}(\underline{s}_{iz} - \underline{b}_z) \right] \quad (\text{B.16}) \end{aligned}$$

$$\begin{aligned}
n_{1,3} = \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{Sidx} \left(\underline{h}_{iy} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) \right. \right. \right. \\
\left. \left. \left. + \underline{h}_{ix} (\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i)) \right) \right) \right. \\
\left. + \underline{\underline{J}}_{Hidx} \left(\underline{h}_{iy} \sin(2\varphi_i) + \underline{h}_{ix} (\cos(2\varphi_i) + 1) \right) \right) \\
\left. + m_{Hi} \underline{h}_{ix} (\underline{h}_{iz} - \underline{b}_z) + m_{Si} \underline{s}_{ix} (\underline{s}_{iz} - \underline{b}_z) \right] \quad (\text{B.17})
\end{aligned}$$

$$\begin{aligned}
n_{2,1} = \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{Sidx} (\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i)) \right. \right. \\
\left. \left. + \underline{\underline{J}}_{Hidx} (\cos(2\varphi_i) - 1) \right) \right. \\
\left. + m_{Hi} (\underline{b}_z - \underline{h}_{iz}) + m_{Si} (\underline{b}_z - \underline{s}_{iz}) \right] \quad (\text{B.18})
\end{aligned}$$

$$n_{2,2} = \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{Sidx} (\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i)) + \underline{\underline{J}}_{Hidx} \sin(2\varphi_i) \right) \right] \quad (\text{B.19})$$

$$\begin{aligned}
n_{2,3} = \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{Sidx} \left(\underline{h}_{iy} (\cos(\varphi_i - \beta_i) - \cos(\varphi_i + \beta_i)) \right. \right. \right. \\
\left. \left. \left. + \underline{h}_{ix} (\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i)) \right) \right) \right. \\
\left. + \underline{\underline{J}}_{Hidx} \left(\underline{h}_{iy} (1 - \cos(2\varphi_i)) + \underline{h}_{ix} \sin(2\varphi_i) \right) \right) \\
\left. + m_{Hi} \underline{h}_{iy} (\underline{h}_{iz} - \underline{b}_z) + m_{Si} \underline{s}_{iy} (\underline{s}_{iz} - \underline{b}_z) \right] \quad (\text{B.20})
\end{aligned}$$

Verallgemeinerte Matrix der Corioliskräfte für die fixierten Koordinaten

$$\mathbf{P} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{1,2} & -p_{1,1} & p_{2,3} \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{B.21})$$

B. Matrizen des Mechanikmodells

$$p_{1,1} = \sum_i \left[-\frac{1}{r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \cos(\varphi_i + \beta_i) + \underline{\underline{J}}_{H_{ixx}} \cos(2\varphi_i) \right) \right] \quad (\text{B.22})$$

$$p_{1,2} = \sum_i \left[-\frac{1}{r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \sin(\varphi_i + \beta_i) + \underline{\underline{J}}_{H_{ixx}} \sin(2\varphi_i) \right) \right] \quad (\text{B.23})$$

$$\begin{aligned} p_{1,3} = \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \left(\underline{h}_{iy} (\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i)) \right. \right. \right. \\ \left. \left. \left. + \underline{h}_{ix} (\sin(\varphi_i - \beta_i) - \sin(\varphi_i + \beta_i)) \right) \right) \right. \\ \left. + \underline{\underline{J}}_{H_{ixx}} \left(\underline{h}_{iy} (\cos(2\varphi_i) - 1) - \underline{h}_{ix} \sin(2\varphi_i) \right) \right) \\ \left. + m_{Hi} \underline{h}_{iy} (\underline{b}_z - \underline{h}_{iz}) + m_{Si} \underline{s}_{iy} (\underline{b}_z - \underline{s}_{iy}) \right] \quad (\text{B.24}) \end{aligned}$$

$$\begin{aligned} p_{2,3} = \sum_i \left[\frac{1}{2r_i} \left(\gamma_i \underline{\underline{J}}_{S_{ixx}} \left(\underline{h}_{iy} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) \right. \right. \right. \\ \left. \left. \left. + \underline{h}_{ix} (\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i)) \right) \right) \right. \\ \left. + \underline{\underline{J}}_{H_{ixx}} \left(\underline{h}_{iy} \sin(2\varphi_i) + \underline{h}_{ix} (\cos(2\varphi_i) + 1) \right) \right) \\ \left. + m_{Hi} \underline{h}_{ix} (\underline{h}_{iz} - \underline{b}_z) + m_{Si} \underline{s}_{ix} (\underline{s}_{iz} - \underline{b}_z) \right] \quad (\text{B.25}) \end{aligned}$$

Verallgemeinerte Matrix der Zwangskräfte für die fixierten Koordinaten

$$\mathbf{R} = \begin{pmatrix} r_{1,1} & \dots & r_{1,n} \\ r_{2,1} & \dots & r_{1,n} \\ 0 & \dots & 0 \end{pmatrix} \text{diag}(k_1, \dots, k_n) + \begin{pmatrix} \underline{h}_{1y} - \underline{b}_y & \dots & \underline{h}_{ny} - \underline{b}_y \\ \underline{b}_x - \underline{h}_{1x} & \dots & \underline{b}_x - \underline{h}_{nx} \\ 1 & \dots & 1 \end{pmatrix} \quad (\text{B.26})$$

$$r_{1,i} = \sin(\varphi_i) (\underline{b}_z - \underline{h}_{iz} + r_i) \quad (\text{B.27})$$

$$r_{2,i} = \cos(\varphi_i) (\underline{h}_{iz} - \underline{b}_z - r_i) \quad (\text{B.28})$$

Verallgemeinerte Matrix der Antriebskräfte für die fixierten Koordinaten

$$\mathbf{T} = \begin{pmatrix} \cos(\varphi_1)(\underline{b}_z - \underline{h}_{1z} + r_1) & \dots & \cos(\varphi_n)(\underline{b}_z - \underline{h}_{nz} + r_n) \\ \sin(\varphi_1)(\underline{b}_z - \underline{h}_{1z} + r_1) & \dots & \sin(\varphi_n)(\underline{b}_z - \underline{h}_{nz} + r_n) \\ 0 & \dots & 0 \end{pmatrix} \quad (\text{B.29})$$

Projektionen der Geschwindigkeiten

$$\mathbf{W} = \begin{pmatrix} \sin(\varphi_1) & -\cos(\varphi_1) & w_{1,3} \\ \vdots & \vdots & \vdots \\ \sin(\varphi_n) & -\cos(\varphi_n) & w_{n,3} \end{pmatrix} \text{diag}(1/r_1, \dots, 1/r_n) \quad (\text{B.30})$$

$$w_{i,3} = -\underline{h}_{iy} \sin(\varphi_i) - \underline{h}_{ix} \cos(\varphi_i) \quad (\text{B.31})$$

$$\mathbf{X} = \begin{pmatrix} \cos(\varphi_1) & \sin(\varphi_1) & \underline{h}_{1x} \sin(\varphi_1) - \underline{h}_{1y} \cos(\varphi_1) \\ \vdots & \vdots & \vdots \\ \cos(\varphi_n) & \sin(\varphi_n) & \underline{h}_{nx} \sin(\varphi_n) - \underline{h}_{ny} \cos(\varphi_n) \end{pmatrix} \quad (\text{B.32})$$

Projektionen der Antriebsmomente

$$\mathbf{U} = \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ \vdots & \vdots & \vdots \\ u_{n,1} & u_{n,2} & u_{n,3} \end{pmatrix} \quad (\text{B.33})$$

$$\begin{aligned} u_{i,1} = & \frac{\underline{J}^{Hixx}}{2r_i} \left((\cos(2\varphi_i) - 1) \sin(\varphi_i) - \sin(2\varphi_i) \cos(\varphi_i) \right) \\ & + \frac{\gamma_i \underline{J}^{Siox}}{2r_i} \left(\sin(\beta_i) (\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i)) \right. \\ & \left. - \cos(\beta_i) (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) \right) \end{aligned} \quad (\text{B.34})$$

B. Matrizen des Mechanikmodells

$$u_{i,2} = \frac{\underline{\underline{J}}^{Hixx}}{2r_i} \left((\cos(2\varphi_i) + 1) \cos(\varphi_i) + \sin(2\varphi_i) \sin(\varphi_i) \right) + \frac{\gamma_i \underline{\underline{J}}^{Siox}}{2r_i} \left(\sin(\beta_i) (\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i)) + \cos(\beta_i) (\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i)) \right) \quad (\text{B.35})$$

$$u_{i,3} = \frac{\underline{\underline{J}}^{Hixx}}{2r_i} \left(h_{iy} \left(\cos(\varphi_i) \sin(2\varphi_i) + \sin(\varphi_i) (1 - \cos(2\varphi_i)) \right) + \underline{\underline{h}}_{ix} \left(\sin(\varphi_i) \sin(2\varphi_i) + \cos(\varphi_i) (\cos(2\varphi_i) + 1) \right) \right) + \frac{\gamma_i \underline{\underline{J}}^{Siox}}{2r_i} \left(\cos(\beta_i) \left(\underline{\underline{h}}_{iy} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) + \underline{\underline{h}}_{ix} (\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i)) \right) + \sin(\beta_i) \left(\underline{\underline{h}}_{iy} (\cos(\varphi_i - \beta_i) - \cos(\varphi_i + \beta_i)) + \underline{\underline{h}}_{ix} (\sin(\varphi_i + \beta_i) - \sin(\varphi_i - \beta_i)) \right) \right) \quad (\text{B.36})$$

$$\mathbf{V} = \begin{pmatrix} v_{1,1} & v_{1,2} & v_{1,3} \\ \vdots & \vdots & \vdots \\ v_{n,1} & v_{n,2} & v_{n,3} \end{pmatrix} \quad (\text{B.37})$$

$$v_{i,1} = -\frac{\underline{\underline{J}}^{Hixx}}{r_i} \left(\sin(2\varphi_i) \sin(\varphi_i) + \cos(2\varphi_i) \cos(\varphi_i) \right) - \frac{\gamma_i \underline{\underline{J}}^{Siox}}{r_i} \left(\sin(\varphi_i + \beta_i) \sin(\beta_i) + \cos(\varphi_i + \beta_i) \cos(\beta_i) \right) \quad (\text{B.38})$$

$$v_{i,2} = -\frac{\underline{\underline{J}}^{Hixx}}{r_i} \left(\sin(2\varphi_i) \cos(\varphi_i) - \cos(2\varphi_i) \sin(\varphi_i) \right) - \frac{\gamma_i \underline{\underline{J}}^{Siox}}{r_i} \left(\sin(\varphi_i + \beta_i) \cos(\beta_i) - \cos(\varphi_i + \beta_i) \sin(\beta_i) \right) \quad (\text{B.39})$$

$$\begin{aligned}
v_{i,3} = & \frac{\tilde{J}^{Hixx}}{2r_i} \left(\underline{h}_{iy} (\sin(\varphi_i) \sin(2\varphi_i) + \cos(\varphi_i) \cos(2\varphi_i) - \cos(\varphi_i)) \right. \\
& + \underline{h}_{ix} (-\cos(\varphi_i) \sin(2\varphi_i) + \sin(\varphi_i) (\cos(2\varphi_i) + 1)) \Big) \\
& + \frac{\gamma_i \tilde{J}^{Siox}}{2r_i} \left(\sin(\beta_i) (\underline{h}_{iy} (\sin(\varphi_i + \beta_i) + \sin(\varphi_i - \beta_i)) \right. \\
& + \underline{h}_{ix} (\cos(\varphi_i + \beta_i) + \cos(\varphi_i - \beta_i))) \\
& + \cos(\beta_i) (\underline{h}_{iy} (\cos(\varphi_i + \beta_i) - \cos(\varphi_i - \beta_i)) \\
& \left. + \underline{h}_{ix} (\sin(\varphi_i - \beta_i) - \sin(\varphi_i + \beta_i))) \right) \quad (\text{B.40})
\end{aligned}$$

C. Matrizen des reduzierten Mechanikmodells

$$\mathfrak{Z}_j = \mathbf{Z}_j \mathbf{R}_j^{-1} \quad (\text{C.1})$$

$$\mathfrak{T}_j = (\mathbf{D} - \mathfrak{Z}_j \mathbf{T})^{-1} \quad (\text{C.2})$$

$$\mathfrak{M}_j = \text{diag}_{1 \leq i \leq 3} (\mu_i) \mathbf{R}_j^{-1} \quad (\text{C.3})$$

$$\mathfrak{R}_j = \text{diag}_{1 \leq i \leq 3} (r_i) \mathfrak{T}_j \quad (\text{C.4})$$

$$\mathfrak{A}_j = \begin{pmatrix} \mathfrak{M}_j(\mathbf{N} + \mathbf{T} \mathfrak{T}_j(\mathbf{B} + \mathfrak{Z}_j \mathbf{N})) - \mathfrak{T}_j(\mathbf{B} + \mathfrak{Z}_j \mathbf{N}) \\ \mathfrak{M}_j(\mathbf{N} + \mathbf{T} \mathfrak{T}_j(\mathbf{B} + \mathfrak{Z}_j \mathbf{N})) + \mathfrak{T}_j(\mathbf{B} + \mathfrak{Z}_j \mathbf{N}) \\ \mathbf{U} + \mathfrak{R}_j(\mathbf{B} + \mathfrak{Z}_j \mathbf{N}) \\ -\mathbf{U} - \mathfrak{R}_j(\mathbf{B} + \mathfrak{Z}_j \mathbf{N}) \end{pmatrix} \quad (\text{C.5})$$

$$\vec{\mathfrak{B}}_j = \begin{pmatrix} \mathfrak{M}_j(\vec{\mathbf{G}} + \mathbf{T} \mathfrak{T}_j \mathfrak{Z}_j \vec{\mathbf{G}}) - \mathfrak{T}_j \mathfrak{Z}_j \vec{\mathbf{G}} \\ \mathfrak{M}_j(\vec{\mathbf{G}} + \mathbf{T} \mathfrak{T}_j \mathfrak{Z}_j \vec{\mathbf{G}}) + \mathfrak{T}_j \mathfrak{Z}_j \vec{\mathbf{G}} \\ \mathfrak{R}_j \mathfrak{Z}_j \vec{\mathbf{G}} \\ -\mathfrak{R}_j \mathfrak{Z}_j \vec{\mathbf{G}} \end{pmatrix} \quad (\text{C.6})$$

$$\mathfrak{C}_j = \begin{pmatrix} \mathfrak{T}_j(\mathfrak{Z}_j \mathbf{P} - \mathbf{C}) - \mathfrak{M}_j(\mathbf{P} + \mathbf{T} \mathfrak{T}_j(\mathfrak{Z}_j \mathbf{P} - \mathbf{C})) \\ -\mathfrak{T}_j(\mathfrak{Z}_j \mathbf{P} - \mathbf{C}) - \mathfrak{M}_j(\mathbf{P} + \mathbf{T} \mathfrak{T}_j(\mathfrak{Z}_j \mathbf{P} - \mathbf{C})) \\ -\mathbf{V} - \mathfrak{R}_j(\mathfrak{Z}_j \mathbf{P} - \mathbf{C}) \\ \mathbf{V} + \mathfrak{R}_j(\mathfrak{Z}_j \mathbf{P} - \mathbf{C}) \end{pmatrix} \quad (\text{C.7})$$

$$\vec{\mathfrak{D}} = (\vec{0}_6^T \quad -\vec{\mathbf{M}}_{\min}^T \quad \vec{\mathbf{M}}_{\max}^T)^T \quad (\text{C.8})$$

Symbolverzeichnis

Symbole des Kapitels 2

ε	Maximalabstand
$\vec{\mu}_C$	Erwartungswert der durch die Stichprobe S_C repräsentierten Klasse C
Σ_C	Kovarianzmatrix der durch die Stichprobe S_C repräsentierten Klasse C
τ	Integrationszeit
Ξ	allgemeine partitionierende Menge
ξ_i	allgemeines partitionierendes Element
A_{C_i}	Anzahl der Bildelemente des Segments C_i
\mathfrak{C}	zu unterscheidende Klassen
C	Cluster, Ballung
\mathcal{C}	Partitionierung als Menge von k -Clustern
c_i	Farbzugehörigkeit eines Bildelements bzw. einer Gruppe verbundener Bildelemente
\mathfrak{D}	allgemeiner Wertebereich
D	allgemeine Eingabedaten als Stichprobe bzw. als Teilmenge des Wertebereichs
d	allgemeiner dekomponierender Abstand
\mathfrak{d}	Elementabstand
\mathbf{D}	Matrix der klassifizierten Bildelemente
d_B	dekomponierender Abstand entsprechend Satz von Bayes

Symbole des Kapitels 2

d_K	Abstand eines Datums zu einem Cluster
d_M	Mahalanobisdistanz
d_P	Abstand zweier klassifizierter Bildelemente
d_R	Abstand zwischen Datum und Muster
d_S	minimaler Stichprobenabstand
E	Erwartungswert
$F_{i,j}$	Farbfilter
\mathcal{G}	Gruppen verbundener Bildelemente des Bildes, zeilenweise organisiert
g	Kamaverstärkung in dB
G_i	Gruppe verbundener Bildelemente in einer Zeile
I	mittlere Bildhelligkeit
k_C	Umfang der Stichprobe S_C
k_S	Strukturparameter
k_τ	Dämpfungsfaktor
L_i	Menge der Gruppen verbundener Bildelemente einer Zeile
\mathcal{M}	Muster
m	Mindestmächtigkeit
\mathcal{M}^*	durch Ausgleichsrechnung aus einer Unterstützermenge bestimmtes Muster
N_ε	Epsilonumgebung
P	Wahrscheinlichkeit
P_i	zufällige Teilmenge der Eingabemenge zur Erzeugung eines Musters
P_i^*	ε -Unterstützer eines Musters
Q	allgemeine zu partitionierende Menge
r_C	Risiko einer Fehlsuordnung zur Klasse C
\vec{r}_{C_i}	Schwerpunkt des Segments C_i

r_i	Bezeichner der Wurzelgruppe eines Segments
S	Struktur
S_C	Stichprobe zur Klasse C
S_i	Stichprobe aus D vom Umfang m_i . Repräsentiert Cluster C_i
s_i	Bezeichner einer Gruppe verbundener Bildelemente bzw. eines Segments
V	bayerkodierte Bildmatrix
\mathcal{V}	Voronoizelle
$v_{i,j}$	Wert eines Bildelements der bayerkodierten Bildmatrix
$\vec{v}_{i,j}$	rekonstruierter Wert eines Bildelements
w_i	Breite einer Gruppe verbundener Bildelemente
x_i	Spalte eines Bildelements bzw. Startspalte einer Gruppe verbundener Bildelemente
y_i	Zeile eines Bildelements

Symbole des Kapitels 3

δ	gewichteter Lagefehler
$\vec{\varepsilon}_i$	Klaffung des identen Punktes i
ω_i	Wichtung der Klaffung i
φ	Drehwinkel des Beobachtersystems
φ_0	Startwinkel eines Kreisringsektors
φ'_0	Startwinkel eines Kreisringsektors, gestört
φ_1	Endwinkel eines Kreisringsektors
φ'_1	Endwinkel eines Kreisringsektors, gestört
A	Flächeninhalt
A'	Flächeninhalt einer gestörten Markierung

Symbole des Kapitels 4

E	Summe der Klaffungen
\vec{E}_i	Einheitsvektoren des Beobachtersystems
\vec{e}_i	Einheitsvektoren des Referenzsystems
m	Abbildungsmaßstab
n	Anzahl identer Punkte
r_R	Radius des Spielers
\vec{r}_0	Verschiebung im Referenzsystem
r_A	Radius der Zentralmarkierung
r_a	Außenradius eines Kreisringsektors
r'_a	Außenradius eines Kreisringsektors, gestört
\vec{R}_i	reduzierte Punkte im Referenzsystem
\vec{r}_i	Ortsvektor auf identen Punkt i im Referenzsystem
r_i	Innenradius eines Kreisringsektors
r'_i	Innenradius eines Kreisringsektors, gestört
\vec{s}_0	Verschiebung im Beobachtersystem
\vec{S}_i	reduzierte Punkte im Beobachtersystem
\vec{s}_i	Ortsvektor auf identen Punkt i im Beobachtersystem
x_0	x-Komponente eines Kreisringsektorzentrums
x'_0	x-Komponente eines Kreisringsektorzentrums, gestört
y_0	y-Komponente eines Kreisringsektorzentrums
y'_0	y-Komponente eines Kreisringsektorzentrums, gestört

Symbole des Kapitels 4

β_i	Orientierung der Antriebsachse des Antriebs i im Roboterkoordinatensystem
ΔT	Abtastzeit

ε	tolerierte Überswingweite
γ_i	Übersetzung des Antriebs i
Λ	Skalierung der Beschleunigung für das globale Koordinatensystem
λ	Skalierung der Beschleunigung für das Roboterkoordinatensystem
Λ_{\min}	kleinste Beschleunigungsskalierung für Trajektorie aus Zustand, Sollzustand und Einstellzeit
μ_{gi}	Rollreibungszahl eines Kranzrades des Rades i
μ_i	Haftreibungszahl des Rades i
ω_i	Winkelgeschwindigkeit des Rades i um die Achse des Rades i
ω_{ri}	Sollwinkelgeschwindigkeit des Rades i um die Achse des Rades i
ω_i^*	gemessene Radwinkelgeschwindigkeit des Rades i
φ_i	Orientierung der Radachse des Rades i im Roboterkoordinatensystem
Ψ	Ausrichtung des Roboters im globalen Koordinatensystem
ρ	stückweises Polynom vom Grad 2 zur Lageregelung
ξ_i	Winkelgeschwindigkeit des Antriebs i um die Antriebsachse des Antriebs i
ζ	Polynom vom Grad 3 zur Lageregelung
\mathfrak{A}_j	Ersatztableau, Beschleunigungsterm des Betriebsfalls j
\vec{A}_{Hi}	Schnittkraft zwischen Grundkörper und Rad i
\vec{A}_{Si}	Schnittkraft zwischen Grundkörper und Antrieb i
\mathbf{B}	verallgemeinerte Trägheitsmatrix der gesteuerten Koordinaten
$\vec{\mathfrak{B}}_j$	Ersatztableau, Konstante des Betriebsfalls j
B	Basis des Roboterkoordinatensystems im globalen Koordinatensystem
\underline{B}_B	Basis des Koordinatensystems des Grundkörpers im Roboterkoordinatensystem

Symbole des Kapitels 4

\underline{B}_{Hi}	Basis des Koordinatensystems des Rades i im Roboterkoordinatensystem
\vec{b}	Ortsvektor auf den Schwerpunkt des Grundkörpers im Roboterkoordinatensystem
\underline{B}_{Si}	Basis des Koordinatensystems des Antriebs i im Roboterkoordinatensystem
\mathbf{C}	verallgemeinerte Matrix der Corioliskräfte für die gesteuerten Koordinaten
\vec{c}_j	Ersatztableau, Coriolisterm des Betriebsfalls j
\mathbf{D}	verallgemeinerte Matrix der Antriebskräfte für die gesteuerten Koordinaten
\vec{d}	Ersatztableau, Geschwindigkeitsterm
\vec{d}	Vektor der Antriebskraftkomponenten
D_i	Haftreibungskraftkomponente des Rades i in \vec{E}_{Hiy}
\vec{e}	Geschwindigkeitsfehler
\vec{e}_i	Einheitsvektoren des globalen Koordinatensystems
$\vec{\underline{E}}_{Bj}$	Einheitsvektoren des Koordinatensystems des Grundkörpers im Koordinatensystem des Grundkörpers
$\vec{\underline{E}}_{Bj}$	Einheitsvektoren des Koordinatensystems des Grundkörpers im Roboterkoordinatensystem
$\vec{\underline{E}}_{Hij}$	Einheitsvektoren des Koordinatensystems des Rades i im Koordinatensystem des Rades i
$\vec{\underline{E}}_{Hij}$	Einheitsvektoren des Koordinatensystems des Rades i im Roboterkoordinatensystem
\vec{E}_i	Einheitsvektoren des Roboterkoordinatensystems im globalen Koordinatensystem
$\vec{\underline{E}}_{Sij}$	Einheitsvektoren des Koordinatensystems des Antriebs i im Koordinatensystem des Antriebs i
$\vec{\underline{E}}_{Sij}$	Einheitsvektoren des Koordinatensystems des Antriebs i im Roboterkoordinatensystem

\vec{F}_B	Summe der am Grundkörper angreifenden Kräfte
\vec{F}_{Hi}	Summe der am Rad i angreifenden Kräfte
\vec{f}_i	Aufstandspunkt des Rades i
\vec{F}_{Si}	Summe der am Antrieb i angreifenden Kräfte
\vec{G}	gemischter Vektor des Einflusses der Erdbeschleunigung
g	Erdbeschleunigung
H_i	Rollreibungskraftkomponente des Rades i in \vec{E}_{Hix}
\vec{h}_i	Ortsvektor auf den Schwerpunkt des Rades i im Roboterkoordinatensystem
I	elektrische Stromstärke
J	Trägheitsmoment des Antriebsstrangs
$\underline{\underline{J}}_{Hixx}$	Trägheitsmoment des Rades i bezüglich Rotation um die Radachse $\underline{\underline{\vec{E}}}_{Hix}$ des Rades i
$\underline{\underline{J}}_{Hiiy}$	Trägheitsmoment des Rades i bezüglich Rotation um $\underline{\underline{\vec{E}}}_{Hiiy}$
$\underline{\underline{J}}_{Hizz}$	Trägheitsmoment des Rades i bezüglich Rotation um $\underline{\underline{\vec{E}}}_{Hizz}$
$\underline{\underline{J}}_{Siox}$	Trägheitsmoment des Antriebs i bezüglich Rotation um die Antriebsachse $\underline{\underline{\vec{E}}}_{Siox}$ des Antriebs i
$\underline{\underline{J}}_{Siiy}$	Trägheitsmoment des Antriebs i bezüglich Rotation um $\underline{\underline{\vec{E}}}_{Siiy}$
$\underline{\underline{J}}_{Sizz}$	Trägheitsmoment des Antriebs i bezüglich Rotation um $\underline{\underline{\vec{E}}}_{Sizz}$
k	Verstärkung der angenommenen Bremsbeschleunigung
k_i	Geschwindigkeitsbereinigte Rollreibungszahl des Rades i
k_m	Maschinenkonstante
k_v	Verstärkung des Geschwindigkeitsreglers
\vec{L}_B	Drehimpuls des Grundkörpers
\vec{L}_{Hi}	Drehimpuls des Rades i

Symbole des Kapitels 4

\vec{L}_{Si}	Drehimpuls des Antriebs i
M	Antriebsmoment in Achsrichtung
m_B	Masse des Grundkörpers
\vec{M}_B	Summe der am Grundkörper angreifenden Momente
\vec{M}_G	Vektor der Antriebsmomentkomponenten, Induktionsspannungsterm
m_{Hi}	Masse des Rades i
\vec{M}_{Hi}	Summe der am Rad i angreifenden Momente
\vec{M}_I	Vektor der Antriebsmomentkomponenten, Maximalstromterm
\vec{M}_{\max}	Vektor der Antriebsmomentkomponenten, komponentenweises Maximum
\vec{M}_{\min}	Vektor der Antriebsmomentkomponenten, komponentenweises Minimum
M_{Li}	Lastmomentkomponente des Rades i in Richtung der Achse des Rades i
\vec{M}_Q	Vektor der Antriebsmomentkomponenten, Klemmspannungsterm
m_{Si}	Masse des Antriebs i
\vec{M}_{Si}	Summe der am Antrieb i angreifenden Momente
\mathbf{N}	verallgemeinerte Trägheitsmatrix der fixierten Koordinaten
\mathbf{P}	verallgemeinerte Matrix der Corioliskräfte für die fixierten Koordinaten
\vec{p}_B	Impuls des Grundkörpers
\vec{p}_{Hi}	Impuls des Rades i
\vec{p}_{Si}	Impuls des Antriebs i
Ω	Zustandsraum
Q	Trajektorie aus Zustand, Sollzustand und Einstellzeit
\vec{q}	Lage

\vec{q}	Zustand
$\dot{\vec{q}}$	Geschwindigkeit
$\ddot{\vec{q}}$	Beschleunigung des Roboters im globalen Koordinatensystem
$\ddot{\vec{q}}_0$	Startbeschleunigung
$\ddot{\vec{q}}_e$	Bremsbeschleunigung
$\ddot{\vec{q}}_{\sim}$	Ins Roboterkoordinatensystem projizierte Beschleunigung des Roboters
$\ddot{\vec{q}}_{\sim r}$	Sollbeschleunigung
$\dot{\vec{Q}}$	gemischte quadratische Geschwindigkeit
$\dot{\vec{q}}_{\sim}$	Geschwindigkeit ins Roboterkoordinatensystem projiziert
$\dot{\vec{q}}_r$	Sollgeschwindigkeit
$\dot{\vec{q}}_{\sim r}$	Sollgeschwindigkeit ins Roboterkoordinatensystem projiziert
Ψ	Orientierung des Roboters im globalen Koordinatensystem
$\dot{\Psi}$	Winkelgeschwindigkeit des Roboters um \vec{e}_z
$\ddot{\Psi}$	Winkelbeschleunigung des Roboters um \vec{e}_z
\vec{q}_r	Solllage
r_x, r_y	Positionskomponenten des Roboters im globalen Koordinatensystem
\dot{r}_x, \dot{r}_y	Geschwindigkeitskomponenten des Roboters im globalen Koordinatensystem
\ddot{r}_x, \ddot{r}_y	Beschleunigungskomponenten des Roboters im globalen Koordinatensystem
\mathbf{R}	verallgemeinerte Matrix der Zwangskräfte für die fixierten Koordinaten
\mathbf{R}^+	Pseudoinverse der verallgemeinerten Matrix der Zwangskräfte für die fixierten Koordinaten
\vec{r}	Ortsvektor auf das Roboterkoordinatensystem im globalen Koordinatensystem
R_a	Ankerwiderstand

Symbole des Kapitels 5

r_i	Radius des Rades i
r_i^*	Abstand des Momentanpols der Abrollbewegung des Rades i von der Nabe des Rades i
\vec{s}_i	Ortsvektor auf den Schwerpunkt des Antriebs i im Roboterkoordinatensystem
\mathbf{T}	verallgemeinerte Matrix die Antriebskräfte für die fixierten Koordinaten
T	Umschaltzeit
T_e	Einstellzeit
\vec{T}_{Hi}	Schnittmoment zwischen Grundkörper und Rad i
\vec{T}_{Si}	Schnittmoment zwischen Grundkörper und Antrieb i
\mathbf{U}	Projektion der Schnittmomente auf die Radachsen, Beschleunigungsterm
U_q	elektrische Klemmspannung
\mathbf{V}	Projektion der Schnittmomente auf die Radachsen, Coriolisterm
\mathbf{W}	Kinematische Verknüpfung aus der in das Roboterkoordinatensystem projizierten Robotergeschwindigkeit und den Radwinkelgeschwindigkeiten
\mathbf{W}^+	Pseudoinverse der kinematischen Verknüpfung der Radwinkelgeschwindigkeiten
\mathbf{X}	Projektion der Robotergeschwindigkeit auf die Radachsen
\mathbf{Z}	verallgemeinerte Matrix der Zwangskräfte für die gesteuerten Koordinaten
Z	Güte
\vec{z}	Vektor der Zwangskraftkomponenten
Z_i	Zwangskraftkomponente des Rades i in \vec{e}_z

Symbole des Kapitels 5

$\vec{\Delta}$	Verschiebung eines Polygoneckpunktes
Γ	Nachfolger- bzw. Expansionsoperator
λ	Spurparameter einer Geraden
λ_A	Spurparameter des Austritts einer Strecke aus einem Polygon
λ_E	Spurparameter des Eintritts einer Strecke in ein Polygon
Φ	normalisierter Winkel
a	Mindestabstand des Roboters von einem Hindernis
C	Zellteilungsoperator
c	Kinder eines Knotens
c_d	Kinder eines Knotens, die an Kante d grenzen
c_{ij}	Transitionskosten von Knoten i zu Knoten j
e_{ij}	Kante von Knoten i zu Knoten j eines Graphen
f	Expansionskriterium
G	Graph
g	Kosten eines Pfades
h	Heuristik
\mathcal{K}	Kante eines Polygons
k	Anzahl der Eckpunkte eines Polygons
\mathcal{K}_N	Nordkante einer Zelle
\mathcal{K}_O	Ostkante einer Zelle
\mathcal{K}_S	Südkante einer Zelle
\mathcal{K}_W	Westkante einer Zelle
L_d	Blätter an Kante d
N	Navigationspunkte
\vec{n}	Normalenvektor einer Polygonkante

Symbole des Kapitels 5

N	Blattnachbarschaft
N_d	Nachbarblätter an Kante d
n_i	Knoten eines Graphen
\vec{o}	Ortsvektor auf Hindernisposition
\mathcal{O}	Hindernis
o	Störung einer Zelle
\mathcal{P}	konvexes Polygon
P	Eckpunkt eines Polygons
\vec{p}	Ortsvektor auf einen Eckpunkt eines Polygons
p	Elter eines Knotens
R	Position des Roboters
\vec{r}	Ortsvektor auf Roboterposition
S	Strecke zwischen Navigationspunkten
T	Zielposition
\vec{u}	Richtungsvektor einer Strecke
\vec{v}	Richtungsvektor einer Polygonkante
W	Wegpunkt
\mathcal{Z}	Zelle eines Quadtree
Z_{NO}	nordöstlicher Eckpunkt einer Zelle
\mathcal{Z}_{NO}	nordöstliche Kindzelle
Z_{NW}	nordwestlicher Eckpunkt einer Zelle
\mathcal{Z}_{NW}	nordwestliche Kindzelle
Z_{SO}	südöstlicher Eckpunkt einer Zelle
\mathcal{Z}_{SO}	südöstliche Kindzelle
Z_{SW}	südwestlicher Eckpunkt einer Zelle
\mathcal{Z}_{SW}	südwestliche Kindzelle

Abbildungsverzeichnis

1.1. Reale Spielumgebung	3
1.2. Modell Lukas[Jah09]	4
1.3. RoboCup Spieler der TU Ilmenau, Modell M11 [Mül09, Sch10]	5
1.4. Softwaremodule des autonomen Fußballspiels	6
1.5. Paradigmen der Verhaltensmodellierung	8
1.6. Verhaltensmodellierung im RoboCup	10
2.1. Kameravarianten	17
2.2. Nächster-Nachbar-Interpolation	18
2.3. Gemischt lineare/bilineare Interpolation	19
2.4. Lineare Interpolation	20
2.5. Helligkeitsverlauf des Spielfeldes	22
2.6. Idealisierte Farbklassen	25
2.7. Streudiagramme der auftretenden Farbwerte im RGB-Farbraum	27
2.8. Minimale einhüllende Quader	28
2.9. Diskrete Voronoidekomposition	33
2.10. Stichprobenzuordnung	34
2.11. Streudiagramme der auftretenden Farbwerte im HSV-Farbraum	40
2.12. Gegenüberstellung von nutzerklassifizierten Stichproben und DBSCAN-Clustern	44
2.13. Vereinfachte Bereitstellung der Eingabedaten durch grobes Mar- kieren	44
2.14. Bisheriges Segmentierungsverfahren	46
2.15. Beispiel der Vereinigung erreichbarer Gruppen	50
3.1. Startmerkmale	52
3.2. Trikotvergleich	54
3.3. Helmert-Transformation	56
3.4. Simulierte Fehlklassifikation	64
3.5. Mittlere Positionsabweichungen	67
3.6. Mittlere Winkelabweichungen	68
3.7. Mittlere normierte Lageabweichungen	69

Abbildungsverzeichnis

4.1. Omnidirektionales Antriebssystem	73
4.2. Kinematische Steuerung	74
4.3. Kinematische Antriebssteuerung	75
4.4. Omnidirektionales Antriebssystem als Mehrkörpersystem . . .	78
4.5. Koordinatensysteme	79
4.6. Kräfte am Omnidirektionalrad	89
4.7. Elektrisches Modell	106
4.8. Beschleunigungsgrenzen regelmäßiger Antriebskonfigurationen	111
4.9. Beschleunigungsgrenzen dreirädriger Antriebskonfigurationen	112
4.10. Zustandsabhängige Beschleunigungsgrenze	113
4.11. Einfluss des Radschlupfs	115
4.12. Geschwindigkeitsregler	119
4.13. Führungssprünge am Geschwindigkeitsregler	120
4.14. Testraster zur Bestimmung der Verstärkung der Bremsbeschleunigung	126
4.15. Führungssprung am Lageregler	129
4.16. Spur des lagegeregelten Roboters bei Führungssprung	130
5.1. Bewegung eines Roboters auf einem Spielfeld mit Hindernissen	131
5.2. Beispiel eines gewichteten und gerichteten zyklischen Graphen.	134
5.3. Beispiel A^*	136
5.4. Konstruktion des Sichtbarkeitsgraphen in der Spielfeldfläche .	137
5.5. Approximation von Hindernissen durch konvexe Polygone . . .	140
5.6. Punkt-im-Polygon-Test	142
5.7. Segment-Polygon-Schnitt	142
5.8. Vertikale Trapezdekomposition	144
5.9. Navigationsgraph aus adaptiver Zelldekomposition	145
5.10. Zelle des Quadtree	145
5.11. Störungen einer Zelle	146
5.12. Varianten der Blattnachbarschaft	147
5.13. Blattnachbarschaft	148
5.14. Pfade bei unterschiedlichen Transitionskosten	152
5.15. Gezieltes Überspringen	156
5.16. Torschuss	157
5.17. Verteidigung	157

Tabellenverzeichnis

2.1. Leistungskennwerte des Segmentierungsverfahrens	49
3.1. Parameter des vereinfachten Markierungsmodells	63
3.2. Grenzgüten beim Einsatz der verschiedenen Gütekriterien . . .	68
4.1. Verallgemeinerte Matrizen des Mechanikmodells	103
4.2. Bestimmung der Beschleunigungsgrenze als lineares Optimie- rungsproblem reeller Basisvariablen	109
4.3. Bestimmung der Beschleunigungsgrenze als lineares Optimie- rungsproblem nichtnegativer Basisvariablen	110
4.4. Modellparameter	110
4.5. Verstärkung der Bremsbeschleunigung	126
5.1. Parameter des Navigationsgraphen aus der Quadtree-dekomposi- tion	153
5.2. Parameter des Sichtbarkeitsgraphen aus Polygonen	154

Liste der Algorithmen

1.	RANSAC	36
2.	ExpandiereCluster	42
3.	DBSCAN	43
4.	Lauf­längen­kodierung	47
5.	Vereinige	48
6.	Segmentierung	49
7.	Störung der Zentralmarke	65
8.	Störung einer Außenmarke	66
9.	Finde kleinste Einstellzeit	123
10.	A*	135
11.	Nachbar	149
12.	Nachbarn	149

Literaturverzeichnis

- [Abr01] ABRAHAM, S.: *Entwicklung eines mobilen Roboters für ein RoboCup Team*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2001. – Diplomarbeit
- [AJ05] ATA, A. A. ; JOHAR, H.: Cubic-Spline Trajectory Planning of a Constrained Flexible Manipulator. In: *ICRA, IEEE*, 2005, S. 3126–3130
- [Bau08] BAUMGARTNER, J.: *Automatische Farbklassifikation zur Anwendung im RoboCup*, Technische Universität Darmstadt, Department of Computer Science, Diplomarbeit, 2008
- [BBBV05] BROWNING, B. ; BRUCE, J. ; BOWLING, M. ; VELOSO, M.: STP: Skills, tactics and plays for multi-robot control in adversarial environments. In: *IEEE Journal of Control and Systems Engineering* 219 (2005), S. 33–52
- [BBD⁺09] BLANK, P. ; BLEIER, M. ; DREXLER, S. ; KALLWIES, J. ; KUGLER, P. ; LAHMANN, D. ; NORDHUS, P. ; RIESS, C. ; SWADZBA, T. ; TULLY, J.: ER-Force Team Description Paper for RoboCup 2009. In: FOUNDATION, RoboCup (Hrsg.): *Proceedings-CD*, 2009
- [BBV00] BRUCE, J. ; BALCH, T. ; VELOSO, M.: Fast and inexpensive color image segmentation for interactive robots. In: *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)* Bd. 3, 2000, S. 2061 – 2066
- [BEF84] BEZDEK, J. C. ; EHRLICH, R. ; FULL, W.: FCM: The fuzzy c-means clustering algorithm. In: *Computers & Geosciences* 10 (1984), Nr. 2-3, S. 191 – 203
- [BEG⁺03] BEHNKE, S. ; EGOROVA, A. ; GLOYE, E. ; ROJAS, R. ; SIMON, M.: Predicting away Robot Control Latency. In: *In RoboCup 2003: Robot Soccer World Cup VII*, 2003, S. 712–719

Literaturverzeichnis

- [Beh11] BEHN, C.: Adaptive Control of Straight Worms without Derivative Measurement. In: *Multibody System Dynamics* 26 (2011), Nr. 3, S. 213–243
- [Bel54] BELLMAN, R.: The theory of dynamic programming. In: *Bulletin of the American Mathematical Society* 60 (1954), Nr. 60, S. 503–515
- [BM01] BRÜDERLIN, B. ; MEIER, B.: *Computergrafik und Geometrisches Modellieren*. Teubner-Verlag, 2001
- [Boo] BOOST C++ Libraries. <http://www.boost.org>
- [Bro90] BROOKS, R. A.: Elephants don't play chess. In: *Robotics and Autonomous Systems* 6 (1990), S. 3–15
- [BS09] BEHN, C. ; STEIGENBERGER, J.: Improved Adaptive Controllers for Sensory Systems — First Attempts. In: AWREJCEWICZ, J. (Hrsg.): *Modeling, Simulation and Control of Nonlinear Engineering Dynamical Systems*. Springer Netherlands, 2009. – ISBN 978–1–4020–8778–3, S. 163–178
- [Büs11] BÜSKENS, Christof: Suboptimale Regelung nichtlinearer Prozesse. In: *Automatisierungstechnik* 59 (2011), Nr. 8, S. 489–501
- [BV02] BRUCE, J. ; VELOSO, M.: Real-Time Randomized Path Planning for Robot Navigation. In: *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '02)* Bd. 3, 2002, S. 2383–2388
- [Car06] CARPIN, S.: Randomized Motion Planning - A Tutorial. In: *International Journal of Robotics and Automation* 21 (2006), Nr. 3, S. 184–196
- [CC99] CHUI, C. K. ; CHEN, G.: *Kalman filtering with real-time applications*. 3. Berlin : Springer, 1999
- [CDI97] CAO, B. ; DODDS, G. I. ; IRWIN, G. W.: Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots. In: *Control Theory and Applications, IEE Proceedings*-144 (1997), Nr. 5, S. 467–475

- [CGD⁺01] CARTER, B. ; GOOD, M. ; DOROHOFF, M. ; LEW, J. ; WILLIAMS II, R. L. ; GALLINA, P.: Mechanical Design and Modeling of an Omnidirectional RoboCup Player. In: *RoboCup AI Conference*. Seattle, WA, Aug. 2001
- [CH67] COVER, T. ; HART, P.: Nearest neighbor pattern classification. 13 (1967), S. 21– 27
- [CLH⁺05] CHOSET, H. ; LYNCH, K. M. ; HUTCHINSON, S. ; KANTOR, G. A. ; BURGARD, W. ; KAVRAKI, L. E. ; THRUN, S.: *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA : MIT Press, 2005
- [cor] *Common Object Request Broker Architecture (CORBA)*. <http://www.omg.org/spec/CORBA/index.htm>
- [cpp03] *ISO/IEC 14882:2003: Programming languages: C++*. 2003. – <http://www.iso.org/iso/en/CatalogueDetailPage.-CatalogueDetail?CSNUMBER=38110>
- [CV05] CAMASTRA, F. ; VERRI, A.: A Novel Kernel Method for Clustering. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (2005), May, S. 801–804. – ISSN 0162–8828
- [DAH00] DUPONT, P. ; ARMSTRONG, B. ; HAYWARD, V.: *Elasto-Plastic Friction Model: Contact Compliance and Stiction*. 2000
- [Dan66] DANTZIG, G. B.: *Lineare Programmierung und Erweiterungen*. Berlin : Springer, 1966 (Ökonometrie und Unternehmensforschung 2)
- [dCvO08] DE BERG, M. ; CHEONG, O. ; VAN KREVELD, M. ; OVERMARS, M.: *Computational Geometry Algorithms and Applications*. 3. Berlin Heidelberg : Springer-Verlag, 2008
- [DH04a] DING, C. ; HE, X.: K-means clustering via principal component analysis. In: *Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA : ACM, 2004 (ICML '04). – ISBN 1–58113–838–5, S. 29–
- [DH04b] DING, C. ; HE, X.: K-Nearest-Neighbor Consistency in Data Clustering: Incorporating Local Information into Global Optimization. In: *Proc. ACM Symposium on Applied Computing (SAC'04)*, 2004, S. 584–589

Literaturverzeichnis

- [Die06] DIESTEL, R.: *Graphentheorie*. Springer Verlag, 2006
- [Dil11] DILL, R.: *Entwicklung eines optischen Messsystems zur Positionsbestimmung von RoboCup-Spielern*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2011. – Bachelorarbeit
- [DM02] DADIOS, E. P. ; MARAVILLAS, Jr. O. A.: Cooperative Mobile Robots with Obstacle and Collision Avoidance Using Fuzzy Logic. In: *International Symposium on Intelligent Control*. Vancouver, Canada : IEEE, October 2002, S. 75–80
- [Eic07] EICHHORN, M.-J.: *Intelligentes Führungskonzept für ein Autonomes Unterwasserfahrzeug in Sondersituationen*, Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, Diss., 2007
- [ESS10] E., Berger ; S., Bähr ; S., Eysten: *Simulation des Fahrverhaltens von mobilen Robotern mit dem Schwerpunkt Radkinematik*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2010. – Projektarbeit
- [Ewa05] EWALD, R.: *Kooperierende Roboter mit Bewegungsbeobachter: Simulation eines Ballwurfes auf ein bewegtes Ziel*. Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, 2005. – Diplomarbeit
- [FB81] FISCHLER, M. A. ; BOLLES, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Commun. ACM* 24 (1981), June, Nr. 6, S. 381–395
- [FDSH97] FOLEY, J. D. ; DAM, A. v. ; STEVEN, S. K. F. ; HUGHES, J. F.: *Computer Graphics: Principles and Practice in C. 2*. Addison Wesley, 1997. – ISBN 0–201–84840–6
- [Föl88] FÖLLINGER, O.: *Optimierung dynamischer Systeme*. 2. München : Oldenbourg, 1988
- [Föl92] FÖLLINGER, O.: *Regelungstechnik*. Heidelberg : Hüthig, 1992
- [FSU11] FRANK, S. ; SCHALE, F. ; USSFELLER, C.: Odometry for mobile robots with laser sensors. In: *56th International Scientific Colloquium*, Technische Universität Ilmenau, 2011

- [Füc06] FÜCHSEL, D.: *Simulation des Fahrverhaltens von mobilen Robotern*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2006. – Diplomarbeit
- [gcc] *GCC, the GNU Compiler Collection*. <http://gcc.gnu.org/>
- [GGA⁺05] GUNTURK, B. K. ; GLOTZBACH, J. ; ALTUNBASAK, Y. ; SCHAFER, R. W. ; MERSEREAU, R. M.: Demosaicking: Color filter array interpolation. In: *IEEE Signal Process. Mag.* 22 (2005), Jan., Nr. 1, S. 44–54
- [GHJV95] GAMMA, E. ; HELM, R. ; JOHNSON, R. ; VLISSIDES, J.: *Design Patterns – Elements of Reusable Object-Oriented Software*. 1. Amsterdam : Addison-Wesley Longman, 1995
- [GJ07] GRUBER, F. J. ; JOCKEL, R.: *Formelsammlung für das Vermessungswesen*. Teubner-Verlag, 2007
- [GM91] GHOSH, S. K. ; MOUNT, D. M.: An Output-Sensitive Algorithm for Computing Visibility Graphs. In: *SIAM Journal of Computing* 20 (1991), Nr. 5, S. 888–910
- [Gra19] GRABOWIECKI, J.: *Vehicle Wheel*. US Patent 1303535, 6 1919
- [GRHHM89] GAL, T. ; R.HORST ; H.ISERMANN ; H.MÜLLER-MERBACH: *Grundlagen des Operations Research*. Bd. 1. 2. Springer Verlag, 1989
- [gtk] *gtkmm C++ Interfaces for GTK+ and GNOME*. <http://www.gtkmm.org/en/>
- [HNR68] HART, P. E. ; NILSSON, N. J. ; RAPHAEL, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In: *Systems Science and Cybernetics, IEEE Transactions on* 4 (1968), Nr. 2, S. 100–107
- [HZ03] HARTLEY, R. I. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, 2003
- [ITM01] IMAI, F. H. ; TSUMURA, N. ; MIYAKE, Y.: Perceptual color difference metric for complex images based on Mahalanobis distance. In: *J. Electronic Imaging* 10 (2001), S. 38539–3

Literaturverzeichnis

- [Jah05] JAHN, M.: *Master-Steuerung für ein Team der Robocup Small-size-League*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2005. – Diplomarbeit
- [Jah09] JAHN, M.: *Ein Beispiel zur Entwicklung kooperierender mobiler Roboter: Konstruktives Design und Steuerungsentwurf*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2009. – Dissertation
- [jav] Java™. <http://www.java.com/>
- [JC98] JAEGER, H. ; CHRISTALLER, T.: Dual dynamics: Designing behavior systems for autonomous robots. In: *Artificial Life and Robotics 2* (1998), S. 76–79
- [JWZ⁺08] JAHN, M. ; WEISS, M. ; ZIMMERMANN, K. ; BRAUNSCHWEIG, M. ; WALKLING, H.-P. ; LERM, S.: RoboCup-Player “LUKAS” – an object of interdisciplinary research and a benchmark problem in mechatronics. In: 53. *Internationales Wissenschaftliches Kolloquium*, 2008
- [Kal60] KALMAN, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME Journal of Basic Engineering D* (1960), Nr. 82, S. 35–45
- [Knu97] KNUTH, Donald E.: *The art of computer programming. Vol. 1, Fundamentals algorithms*. 3. Addison-Wesley Pub. Co., 1997
- [KO05] KLIMEŠOVÁ, D. ; OCELÍKOVÁ, E.: Bayes classifier in multidimensional data classification. In: *15th International Conference on Process Control 2005 in Štrbské Pleso, Slovakia*, 2005, S. 188–1 – 188–5
- [Lan05] LANDSMANN, P.: *Analyse und Weiterentwicklung der Mikrocontrollersoftware für den Spieler der Robocup Smallsize League*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2005. – Projektarbeit
- [LARP98] LARSEN, T. D. ; ANDERSEN, N. A. ; RAVN, O. ; POULSEN, N. K.: Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter. In: *In Proceedings of the 37th IEEE Conference on Decision & Control*, 1998, S. 3972–3977

- [LaV06] *Kapitel Sampling-Based Motion Planning.* In: LAVALLE, S.-M.: *Planning Algorithms.* Cambridge University Press, 2006, S. 183–245

- [LBG03] LINDE, Y. ; BUZO, A. ; GRAY, R.: An Algorithm for Vector Quantizer Design. In: *Communications, IEEE Transactions on* 28 (2003), January, Nr. 1, S. 84–95

- [Ler08] LERM, S.: *Entwurf und Implementation einer neuen Version der Bildverarbeitung für die RoboCup Small Size League.* Technische Universität Ilmenau, Fakultät für Maschinenbau, 2008. – Diplomarbeit

- [Lin94] LIN, J.: An Efficient Parallel Algorithm for Vector Quantizer Design. In: *Proceedings of the 1994 International Conference on Parallel Processing - Volume 03.* Washington, DC, USA : IEEE Computer Society, 1994 (ICPP '94). – ISBN 0-8493-2493-9, S. 25–30

- [LK99] LAVALLE, S.-M. ; KU, J.: Randomized kinodynamic planning. In: *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1999. – To appear.

- [Lun08] LUNZE, J.: *Regelungstechnik 2: Mehrgrößensysteme, digitale Regelung*. 5. Berlin : Springer, 2008

- [LW95] LI, C.-T. ; WILSON, R.: Image Segmentation Using Multiresolution Fourier Transform / Department of Computer Science, University of Warwick, Coventry. 1995. – Forschungsbericht

- [LZWW08] LIU, Y. ; ZHU, J. J. ; WILLIAMS II, R. L. ; WU, J.: Omnidirectional mobile robot controller based on trajectory linearization. In: *Robotics and Autonomous Systems* 56 (2008), S. 461–479

- [M. 96] M. ESTER AND H.-P. KRIEGEL AND J. SANDER AND X.XU: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, AAAI Press, 1996, S. 226–231

- [Mac67] MACQUEEN, J. B.: Some Methods for Classification and Analysis of MultiVariate Observations. In: CAM, L. M. L. (Hrsg.) ; NEYMAN, J. (Hrsg.): *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* Bd. 1, University of California Press, 1967, S. 281–297

Literaturverzeichnis

- [Mah36] MAHALANOBIS, P. C.: On the generalised distance in statistics. In: *Proceedings National Institute of Science, India* Bd. 2, 1936, S. 49–55
- [Mar07] MARTENS, T.: *Analyse von verschiedenen numerischen Lösungsverfahren für gewöhnliche Differentialgleichungen zur Behandlung von Coulombscher Reibung in mechanischen Bewegungssystemen*. Technische Universität Ilmenau, Fakultät für Maschinenbau, Fachgebiet Technische Mechanik, 2007. – Projektarbeit
- [max09] *Maxima, a Computer Algebra System*. <http://maxima.sourceforge.net/>. Version: 2009. – Version 5.18.1
- [MDSG02] MESSOM, C. H. ; DEMIDENKO, S. ; SUBRAMANIAM, K. ; GUPTA, G. S.: Size/position identification in real-time image processing using run length encoding. In: *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE* Bd. 2, 2002. – ISSN 1091–5281, S. 1055 – 1059 vol.2
- [MRW68] MARTIN, R. S. ; REINSCH, C. ; WILKINSON, J. H.: Householder's tridiagonalization of a symmetric matrix. In: *Numerische Mathematik* 11 (1968), 3, Nr. 3, S. 181–195
- [MTJAL05] MICHELI-TZANAKOU, E. ; JIN, Y. ; ANGELINI, E. ; LAINE, A.: Wavelets in Medical Image Processing: Denoising, Segmentation, and Registration. In: SURI, J. S. (Hrsg.) ; WILSON, D. L. (Hrsg.) ; LAXMINARAYAN, S. (Hrsg.): *Handbook of Biomedical Image Analysis*. Springer US, 2005 (Topics in Biomedical Engineering. International Book Series). – ISBN 978–0–306–48551–0, S. 305–358
- [Mül09] MÜLLER, A.: *Antriebskonzept eines RoboCup Spielers*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2009. – Bachelorarbeit
- [Nil69] NILSSON, N. J.: A Mobile Automation: an Application of Artificial Intelligence Techniques. In: *International Joint Conference on Artificial Intelligence, 7-9 May 1969, Washington, DC.*, 1969, S. 509–520

- [OEW99] OATES, T. ; EYLER-WALKER, Z. ; COHEN, P. R.: Using Syntax to Learn Semantics: An Experiment in Language Acquisition with a Mobile Robot / University of Massachusetts at Amherst. 1999. – Forschungsbericht
- [omn] *omniORB* : *Free CORBA ORB*.
<http://omniorb.sourceforge.net/>
- [Pea01] PEARSON, K.: On lines and planes of closest fit to systems of points in space. In: *Philosophical Magazine* 2 (1901), Nr. 6, S. 559–572
- [Pet10] PETZOLD, N.: *Ansteuerung eines Brushless DC-Motors als Radantrieb eines Spielers der Robocup Small Size League mit einem Mikrocontroller XC888*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2010. – Bachelorarbeit
- [pro] AVT *PvAPI SDK für GigE Vision® Kameras*. – <http://www.alliedvisiontec.com/de/produkte/software/linux/-gige-linux-sdk.html>
- [PTVF92] PRESS, W. H. ; TEUKOLSKY, S. A. ; VETTERLING, W. T. ; FLANNERY, B. P.: *Numerical recipes in C (2nd ed.): the art of scientific computing*. New York, NY, USA : Cambridge University Press, 1992. – ISBN 0–521–43108–5
- [Ren09] RENKA, R. J.: Image segmentation with a Sobolev gradient method. In: *Nonlinear Analysis Theory Methods Applications* 71 (2009), Nr. 12, S. e774–e780
- [RF06] ROJAS, R. ; FÖRSTER, A. G.: Holonomic Control of a robot with an omnidirectional drive. In: *KI - Künstliche Intelligenz* 20 (2006), Nr. 2, S. 12–17
- [RMS91] RITTER, H. ; MARTINETZ, T. ; SCHULTEN, K.: *Neuronale Netze: Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke*. 2. Bonn : Addison-Wesley, 1991
- [RN95] RUSSELL, S. J. ; NORVIG, P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995
- [rob] *RoboCup*. <http://www.robocup.org/>

Literaturverzeichnis

- [Röl00] RÖLLICH, T.: *RoboCup: Gesamtkonzeption, Elektronik und Bilderkennung*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2000. – Diplomarbeit
- [SB98] SUTTON, R. S. ; BARTO, A. G.: *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 1998
- [Sch09] SCHALE, F.: *RoboCup - Schussmechanismen*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2009. – Bachelorarbeit
- [Sch10] SCHALE, F.: *Implementation von Kommutierungs- und Regelungs-algorithmen für elektronisch kommutierte Gleichstrommotoren auf Mikrocontrollern*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2010. – Masterarbeit
- [Sch11] SCHWACHENWALDE, S.: *Inbetriebnahme einer Embedded PC Plattform mit Windows Ce6.0 und Entwicklung einer Demoapplikation mittels Visual Studio*. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2011. – Bachelorarbeit
- [SPK⁺07] SRISABYE, J. ; PARKPIEN, N. ; KONGNIRATSIKUL, P. ; HO-ONSUWAN, P. ; BOWARNKITIWONG, S. ; ARCHAWANANT-HAKUL, M. ; DUMNERNKITTIKUL, R. ; CHONGKAONAR, S. ; RATANAPARADORN, A. ; KEAWPROMMAN, C. ; LIMNIRUN-KUL, V. ; TIPSUWAN, Y.: *Skuba 2007 Team Description*. 2007
- [Str00] STROUSTRUP, B.: *The C++ Programming Language*. 3rd. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2000
- [Sud06] SUD, A.: *Efficient computation of discrete voronoi diagram and homotopy-preserving simplified medial axis of a three-dimensional polyhedron*. University of North Carolina at Chapel Hill Chapel Hill, NC, USA, 2006. – Dissertation
- [Sun] SUNDAY, D.: *Intersection of a Segment with a Convex Polygon or Polyhedron*. http://softsurfer.com/Archive/algorithm_0111/algorithm_0111.htm, . – besucht 2011
- [SV97] STEELS, L. ; VOGT, P.: *Grounding Adaptive Language Games in Robotic Agents*. In: *Proceedings of the Fourth European Conference on Artificial Life*, MIT Press, 1997, S. 474–482

- [TBF05] THRUN, S. ; BURGARD, W. ; FOX, D.: *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, Massachusetts : The MIT Press, 2005
- [Top99] TOPOR, A.: *Roboterfußball: Pfadplanung in dynamischer Umgebung*. Universität Freiburg, Institut für Informatik, 1999. – Diplomarbeit
- [TP97] THRUN, S. ; PRATT, L. ; THRUN, S. (Hrsg.) ; PRATT, L. (Hrsg.): *Learning To Learn*. Kluwer Academic Publishers, 1997
- [Ußf07] USSFELLER, C.: *Entwurf einer Bibliothek zur Steuerung einer Mannschaft der Robocup-Smallsize-League*. Technische Universität Ilmenau, Fachgebiet Rechneranwendung im Maschinenbau, 2007. – Projektarbeit
- [Ußf08] USSFELLER, C.: *Entwurf der Softwarestruktur und Realisierung einzelner Module zur Steuerung einer Mannschaft der Robocup Small Size League*. Technische Universität Ilmenau, Fachgebiet Rechneranwendung im Maschinenbau, 2008. – Diplomarbeit
- [Ußf11] USSFELLER, C.: Application of image processing for autonomous robots. In: *56th International Scientific Colloquium*, Technische Universität Ilmenau, 2011
- [VMM09] VELIĆ, M. ; MAY, D. ; MORESI, L.: A Fast Robust Algorithm for Computing Discrete Voronoi Diagrams. In: *Journal of Mathematical Modelling and Algorithms* 8 (2009), S. 343–355. – ISSN 1570–1166
- [Vol03] VOLK, W.: *Mathematischer Exkurs II: Die Helmert-Transformation*. <http://www.w-volk.de/museum/mathex02.htm>, 6 2003. – besucht 2011
- [WCGR02] WILLIAMS II, R. L. ; CARTER, B. E. ; GALLINA, P. ; ROSATI, G.: Dynamic model with slip for wheeled omnidirectional robots. In: *Robotics and Automation, IEEE Transactions on* 18 (2002), Nr. 3, S. 285–293
- [Wei98] WEIGEL, T.: *Roboter-Fußball: Selbstlokalisierung, Weltmodellierung, Pfadplanung und verhaltensbasierte Kontrolle*. Universität Freiburg, Fakultät für Angewandte Wissenschaften, 1998. – Diplomarbeit

Literaturverzeichnis

- [YGZ03] YE, Q. ; GAO, W. ; ZENG, W.: Color image segmentation using density-based clustering. In: *Multimedia and Expo, IEEE International Conference on* 2 (2003), S. 401–404. ISBN 0–7803–7965–9
- [YSL09] YAN, Y. ; SHEN, Y. ; LI, S.: Unsupervised Color-Texture Image Segmentation Based on A New Clustering Method. In: *Proceedings of the 2009 International Conference on New Trends in Information and Service Science*. Washington, DC, USA : IEEE Computer Society, 2009. – ISBN 978–0–7695–3687–3, S. 784–787
- [Zha00] ZHANG, G. P.: Neural networks for classification: a survey. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 30 (2000), Nr. 4, S. 451–462
- [ZWBR07] ZIMMERMANN, K. ; WEISS, M. ; BRAUNSCHWEIG, M. ; RIESS, T.: *Entwurf einer bahngeführten Positionierung als Basis für die Lokomotion omnidirektionaler Roboter*. Autonome Mobile Systeme, Fachtagung Kaiserslautern, 2007
- [ZZB09] ZIMMERMANN, K. ; ZEIDIS, I. ; BEHN, C.: *Mechanics of Terrestrial Locomotion: With a Focus on Non-pedal Motion Systems*. 1. Heidelberg : Springer, 2009

